



**Universidad Carlos III de Madrid**

Escuela Politécnica Superior  
Grado en Ingeniería Informática  
Especialidad en Computación

# DESARROLLO DE TERAPIAS DE REHABILITACIÓN MOTORA TELEOPERADAS CON EL ROBOT NAO

de  
Trabajo Fin de Grado  
por

Adrián Rossignoli Martínez-Vara de Rey

22 de Junio de 2015

Tutor:

Fernando Fernández Rebollo

Cotutor:

José Carlos González Dorado

# *Abstract*

This bachelor thesis presents two components that recognize the body of the patient through a 3D sensor and give to the therapist and the medical group an interface to increase the knowledge of the system database by adding new poses and exercises for the planning component. It is part of a project called NAOTherapist whose objective is to supervise upper-limb rehabilitation sessions for kids with Obstetric Brachial Plexus Palsy and Cerebral Palsy. This project was developed to help the therapists with their sessions which require continuous attention from the patients, and so that it would also let them treat more patients. If a robot plans the therapy session, checks how the patient is doing and corrects him, the rehabilitation session can be managed autonomously with the robot alone.

This bachelor thesis consists on recognizing the body of the patient through the 3D sensor creating a complete model of the body, showing a skeleton visor, and facilitates the capture of new poses and exercises to improve the basic knowledge of the robot so that it performs better in the therapeutic sessions.

# *Resumen*

Este trabajo está compuesto por dos componentes, uno que reconoce el cuerpo del paciente a través de un sensor 3D y otro que proporciona al terapeuta y/o médico rehabilitador una interfaz para añadir nuevas poses y ejercicios a la base de conocimiento del sistema. Forma parte del proyecto NAOTherapist, cuyo objetivo es la supervisión de sesiones de rehabilitación de los miembros superiores para niños con Parálisis Braquial Obstétrica y Parálisis Cerebral. Es un proyecto que nació para ayudar a los terapeutas en sus sesiones que requerían mucha atención y así que éste tenga más tiempo para otros pacientes. Si un robot realiza la planificación de la sesión terapéutica, comprueba como el paciente realiza la sesión y lo corrige, proporciona a los terapeutas más tiempo para otros pacientes, y al mismo tiempo la terapia robótica sirve para que los niños se sientan mucho más motivados en sus sesiones de rehabilitación.

Este proyecto fin de grado reconoce el cuerpo del paciente a través de un sensor 3D, crea un modelo completo del cuerpo humano, muestra un visor del esqueleto y permite la captura de las nuevas poses y ejercicios para mejorar la base de conocimiento del robot y que éste produzca mejores sesiones terapéuticas.

# Índice

Índice .....	3
Índice de Figuras .....	6
Índice de Tablas.....	9
1.- Introducción .....	10
1.1.- Motivación .....	11
1.1.1.- Robótica terapéutica .....	11
1.1.2.- Reconocimiento de poses.....	14
1.1.3.- Sistema Low-High.....	15
1.1.4.- Visión artificial .....	15
1.2.- Objetivos .....	17
2.- Estado de la cuestión.....	18
2.1.- NAOTherapist.....	18
2.2.- Historia .....	21
2.3.- Kinect .....	30
2.4.- RoboComp .....	35
2.4.1.- ZeroICE .....	35
2.4.2.- WinKinectComp .....	35
3.- Desarrollo .....	41
3.1.- Análisis.....	41
3.1.1.- Escenarios .....	42
3.1.2.- Casos de Uso .....	45
3.1.3.- Requisitos funcionales .....	46
3.2.- Diseño e implementación .....	48
3.2.1.- Herramientas y tecnologías aplicadas.....	48
3.2.2.- Estructura del proyecto .....	49
3.2.3.- Funcionamiento .....	49
3.2.4.- Modelo de VisionComp .....	50
3.2.4.1.- Estructura del modelo.....	51
3.2.4.2.- Modificaciones del modelo .....	52
3.2.5.- VisionComp .....	55
3.2.5.1.- Partes.....	56

3.2.5.1.1.- Body .....	56
3.2.5.1.2.- Hands .....	56
3.2.5.1.3.- Face.....	57
3.2.5.1.4.- Voice.....	57
3.2.5.2.- Imagen en tiempo real.....	57
3.2.5.2.1.- Normalización .....	58
3.2.5.2.2.- Pruebas previas del tratamiento de la imagen.....	59
3.2.6.- LearningComp .....	59
3.2.6.1.- Interfaz gráfica.....	59
3.2.6.2.- Captura de poses .....	61
3.2.6.3.- Captura de ejercicios.....	62
3.2.6.4.- Comparación de poses.....	64
3.2.7.- Estructura de la base de conocimiento .....	64
3.2.7.1.- ChainsL y ChainsR.....	66
3.2.7.3.- ChainsJoint.....	67
3.2.7.4.- Exercises.....	68
4.- Evaluación .....	70
4.1.- Sistema de pruebas.....	70
4.2.- Pruebas.....	71
4.2.1.- Pruebas del componente VisionComp .....	71
4.2.1.1.- Funcionamiento del componente.....	71
4.2.1.2.- Comprobación de los ángulos. ....	73
4.2.2.- Pruebas del componente LearningComp .....	74
4.2.2.1.- Grabar una pose .....	74
4.2.2.2.- Grabar un ejercicio .....	76
5.- Conclusiones .....	78
5.1.- Futuros trabajos .....	79
6.- Referencias y Bibliografía .....	80
7.- Anexo A: Modelos.....	82
7.1.- Modelo de WinKinectComp .....	82
7.1.1.- Módulo RoboCompMSKBody .....	82
7.1.2.- Módulo RobocompMSKRGBD .....	84
7.1.3.- Módulo RoboCompMSKFace.....	85

7.1.4.- Módulo RoboCompMSKASR .....	87
7.1.5.- Módulo RoboCompMSKHand .....	88
7.2.- Modelo de VisionComp .....	89
7.2.1.- Módulo RoboCompKinectModel.....	89
8.- Anexo B: Planificación, presupuesto y marco legal.....	92
8.1.- Planificación .....	92
8.2.- Presupuesto .....	95
8.3.- Marco legal.....	97

# Índice de Figuras

Figura 1. Relación Chains - Pose - Ejercicios - Sesiones – Terapias. ....	10
Figura 2. Exoesqueleto Atlas (García <i>et al</i> , 2011) .....	11
Figura 3. Interacción Humano-Robot. ....	12
Figura 4. Robot Ursus. ....	12
Figura 5. Nueva versión del Robot Ursus.....	13
Figura 6. Campos relacionados con la visión artificial. ....	16
Figura 7. Interfaz del médico para diseñar las terapias de NAOTherapist. ....	19
Figura 8. Diagrama de la arquitectura de NAOTherapist.....	20
Figura 9. Interfaz de grabación de YouMove.....	21
Figura 10. Menú de entrenamiento de YouMove.....	22
Figura 11. Grabación de poses con YouMove.....	23
Figura 12. Comparación de poses con YouMove.....	23
Figura 13. Diagrama de la arquitectura de teleoperación (Alfaro, 2012).....	24
Figura 14. Cálculo de ángulos (Alfaro, 2012). ....	25
Figura 15. Amplitud del hombro (Alfaro, 2012).....	25
Figura 16. Rotación del hombro (Alfaro, 2012).....	26
Figura 17. Amplitud del codo (Alfaro, 2012). ....	26
Figura 18. Codo cerrado (Alfaro, 2012). ....	26
Figura 19. Codo abierto (Alfaro, 2012). ....	27
Figura 20. Fórmula de la rotación del codo (Alfaro, 2012).....	27
Figura 21. Ejemplo de rotación del codo (Alfaro, 2012).....	27
Figura 22. Reconocimiento de pose (O'Hara, 2012).....	28
Figura 23. Diagrama del modelo (Gonzalez-Pacheco <i>et al</i> , 2013).....	29
Figura 24. Detección de pose (Bednarik & Herman, 2015). ....	29
Figura 25. Jugadores de la Xbox con Kinect. ....	30
Figura 26. Cámara de profundidad de la Kinect. ....	31
Figura 27. Funcionamiento de la cámara de profundidad de la Kinect. ....	32
Figura 28. Componentes de la Kinect.. ....	32
Figura 29. Articulaciones del cuerpo humano y el esqueleto para la Kinect. ....	33
Figura 30. Imagen RGB y Profundidad de la Kinect. ....	33
Figura 31. Reconocimiento de la mano. ....	34
Figura 32. Reconocimiento de la cara. ....	34

Figura 33. Diagrama de IceStorm. ....	36
Figura 34. Módulo RoboCompMSKBody del modelo WinKinectComp. ....	37
Figura 35. Módulo RoboCompMSKFace del modelo WinKinectComp. ....	38
Figura 36. Módulo RoboCompMSKRGBD del modelo WinKinectComp. ....	39
Figura 37. Módulo RoboCompMSKHands del modelo WinKinectComp. ....	39
Figura 38. Módulo RoboCompMSKASR del modelo WinKinectComp. ....	40
Figura 39. Caso de uso del escenario E1. ....	45
Figura 40. Caso de uso del escenario E2 y E3. ....	45
Figura 41. Relación Kinect - VisionComp – NAOTherapist. ....	50
Figura 42. Modelo RoboCompKinectModel. ....	51
Figura 43. Clase TPerson: antigua (izquierda) y nueva (derecha). ....	52
Figura 44. Interfaces del Modelo RoboCompKinectModel. ....	53
Figura 45. Código añadido por VisionComp al modelo de WinKinectComp. ....	54
Figura 46. Planos del individuo y sus articulaciones y huesos. ....	55
Figura 47. Cálculo de la apertura de la mano. ....	57
Figura 48. Proyección de la malla de puntos de la cara. ....	57
Figura 49. Reconstrucción del cuerpo con Pyglet. ....	58
Figura 50. Reconstrucción del cuerpo normalizado. ....	59
Figura 51. Interfaz gráfica del LearningComp. ....	60
Figura 52. Captura de nueva pose. ....	61
Figura 53. LearningComp grabando una sesión. ....	62
Figura 54. LearningComp con una sesión generada. ....	63
Figura 55. Botón Show Pose del LearningComp. ....	63
Figura 56. Comparación de poses. ....	64
Figura 57. Almacenamiento de los ángulos de cada brazo en la base de conocimiento. ....	65
Figura 58. Almacenar posición de cada articulación del cuerpo en la base de conocimiento. ...	65
Figura 59. Almacenamiento de los ejercicios en la base de conocimiento. ....	66
Figura 60. Estructura ChainsL. ....	67
Figura 61. Estructura ChainsR. ....	67
Figura 62. Estructura de ChainsJoint. ....	68
Figura 63. Estructura del fichero Exercises de la base de conocimiento. ....	69
Figura 64. Prueba 1 de VisionComp: Postura 1. ....	72
Figura 65. Prueba 1 de VisionComp: Postura 2. ....	72
Figura 66. Prueba 2 de VisionComp: Ángulo 45°. ....	73

Figura 67. Prueba 2 de VisionComp: Ángulo 90°.	74
Figura 68. Prueba 1 de LearningComp: Capturar pose.	75
Figura 69. Prueba 1 de LearningComp: ChainsL.	75
Figura 70. Prueba 1 de LearningComp: ChainsR.	76
Figura 71. Prueba 1 de LearningComp: ChainsJoint.	76
Figura 72. Prueba 2 de LearningComp: Grabar ejercicio.	77
Figura 73. Prueba 2 de LearningComp: Exercises.	77
Figura 74. Planificación inicial.	93
Figura 75. Planificación final.	93
Figura 76. Planificación inicial vs planificación final.	94



# *Índice de Tablas*

Tabla 1. Presupuesto del personal.....	95
Tabla 2. Presupuesto del equipo.....	95
Tabla 3. Presupuesto del software.....	96
Tabla 4. Presupuesto de otros gastos.....	96
Tabla 5. Resumen del presupuesto.....	97

# 1.- Introducción

En este capítulo se introduce el contexto que enmarca al proyecto NAOTherapist del cual este trabajo forma parte. También se explican los protocolos terapéuticos seguidos y se motiva cada una de las partes desarrolladas. Finalmente, se recogen de forma concisa los objetivos concretos de este trabajo.

El proyecto NAOTherapist tiene como objetivo la supervisión de sesiones de rehabilitación de los miembros superiores para niños con Parálisis Braquial Obstétrica y Parálisis Cerebral. El objetivo del proyecto es el reconocimiento de poses mediante un sensor 3D y realizar una interfaz gráfica para que el terapeuta pueda añadir nuevas poses y ejercicios a la base de conocimiento usando su propio cuerpo como modelo para definirlos. El diseño e implementación para la realización del proyecto se dividió en tres niveles de planificación:

- Alto nivel: El diseño de las sesiones de rehabilitación que compondrán la terapia completa (Pulido *et al*, 2014).
- Medio nivel: La planificación de la sesión concreta según los ejercicios previamente planificados. El diseño de la base de conocimiento de poses y ejercicios es retrocompatible con la planificación previa de alto nivel de este proyecto (Pulido, 2014).
- Bajo nivel: El control de los componentes del robot terapeuta (González, 2014).

Este trabajo fin de grado está relacionado con el control de componentes del robot terapeuta, el modelado de la información y la ampliación de la base del conocimiento para el planificador automático de alto nivel.

Para la correcta modelización del problema se ha tenido en cuenta cómo están compuestos los ejercicios de rehabilitación. Todas las terapias que se planifiquen tienen que estar compuestas por una serie de sesiones, éstas a su vez por ejercicios que estarán formados por poses y cada pose estará formada por 2 chains, una para cada brazo (Figura 1).

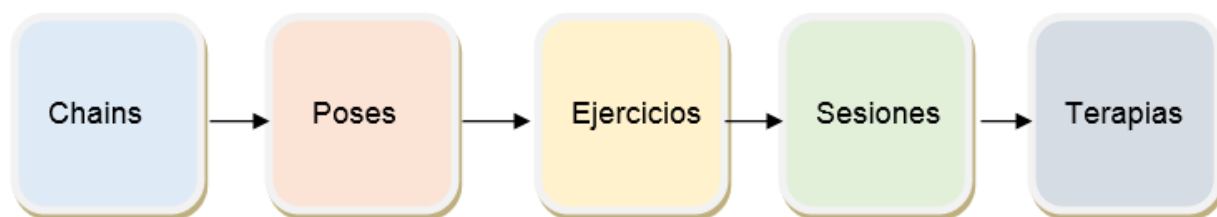


Figura 1. Relación Chains - Pose - Ejercicios - Sesiones – Terapias.

## 1.1.- Motivación

En esta sección del trabajo se introduce el interés actual que existe sobre la robótica terapéutica y el reconocimiento de poses, el cual ha llevado a la realización de este proyecto fin de grado.

### 1.1.1.- Robótica terapéutica

En los últimos años el interés en la robótica terapéutica ha aumentado de forma considerable debido a que se ha visto que los robots son una gran herramienta en este campo como consecuencia de los nuevos avances que pueden aportar en esta área. Este proyecto está orientado a la rehabilitación de niños, los cuales durante las largas sesiones con los terapeutas realizando ejercicios repetitivos, a veces acaban perdiendo el interés por éstos, lo que conlleva un aumento en el tiempo de recuperación del paciente y una mala curación de la parte afectada (Calderita, 2013). Los terapeutas gastan muchos esfuerzos en mantener la atención de los pacientes, y especialmente, en que se esfuercen en hacer sus ejercicios de rehabilitación. Esto produce un gasto de recursos para el paciente y para el hospital.

Por ello se ha investigado mucho en el uso de robots como terapeutas, ya que aportan una motivación extra al paciente a la vez que liberan de trabajo al terapeuta, permitiendo que ambos puedan realizar mejor su tarea.

En la actualidad, existen dos tipos de robots terapéuticos: con contacto físico y sin contacto físico. Los primeros son robots que ayudan a los pacientes de una forma física, tales como exoesqueletos o prótesis robóticas (Figura 2), que ayudan a los pacientes a realizar una serie de ejercicios que no podrían realizar sin este tipo de herramientas. Hay muchos investigadores que han optado por utilizar la robótica terapéutica desde este punto de vista más físico (Mariusz *et al*, 2013), (Castelli, 2011) o (García *et al*, 2011).

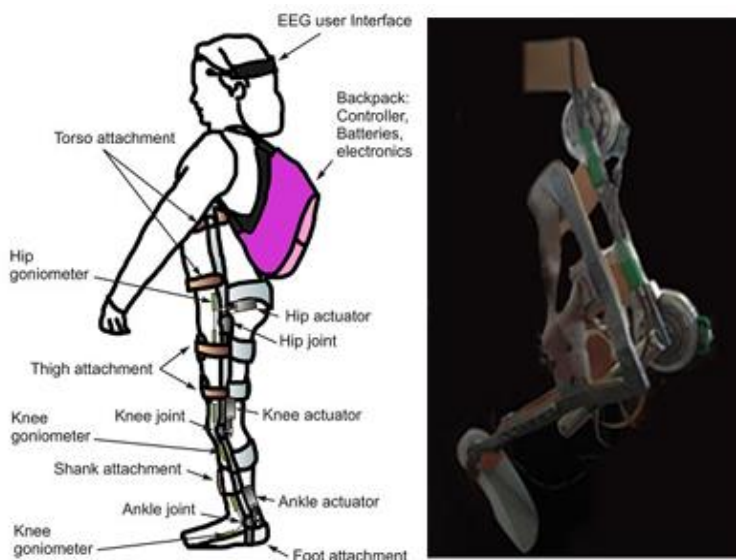
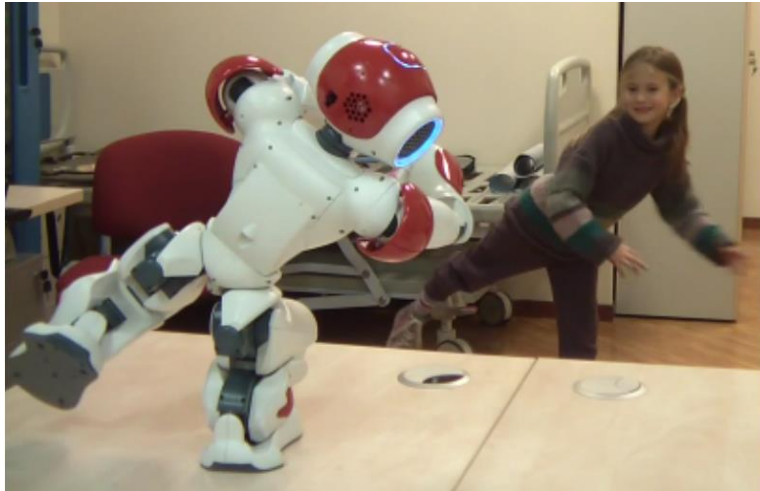


Figura 2. Exoesqueleto Atlas (García *et al*, 2011)

Otros investigadores, en cambio, optan por realizar estudios en los cuales se utiliza la robótica como elemento motivador o como asistente de las funciones del terapeuta (Figura 3). El éxito de estas técnicas reside en la motivación del paciente por parte del robot, creando un vínculo entre el robot y el paciente a la vez que enfocan los tratamientos realizando ejercicios de imitación y de comunicación con el robot. Existen estudios en los que se comprueba como los pacientes obtienen mejores resultados con la interacción con robots (Raquel *et al*, 2011).



*Figura 3. Interacción Humano-Robot.*

El proyecto Ursus abordaba desde este último punto de vista la investigación, la cual, consistió en la construcción de un robot con forma de oso llamado Ursus (Figura 4). Este proyecto con la ayuda de un sensor 3D y la estructura modular RoboComp realiza una sesión de rehabilitación, ejecuta un ejercicio para que el paciente lo imite y luego el robot es teleoperado para mostrar al paciente lo que ha hecho mal y así poder corregirlo (Calderita, 2013).



*Figura 4. Robot Ursus.*

El proyecto demostró cómo los pacientes realizan mejor las sesiones si son guiadas por un robot, pero debido a que este robot no disponía de suficiente autonomía para tomar decisiones, no se reducía el tiempo del terapeuta para que pudiese dedicarlo a otros pacientes o prestar más atención a aspectos clínicos de la rehabilitación, ya que tenía que teleoperar el robot. A causa de las limitaciones encontradas al teleoperar el robot Ursus, se decide expandir este proyecto con Therapist, el cual dota al robot de autonomía para decidir y así mejorar la eficiencia de los terapeutas. La plataforma robótica utilizada en este proyecto es una nueva versión de Ursus (Figura 5) (Calderita, 2013).



*Figura 5. Nueva versión del Robot Ursus.*

Estos proyectos no buscan sustituir a los médicos rehabilitadores ni a los terapeutas, si no que la autonomía del robot sirve para que el uso de éste no entorpezca la labor del terapeuta, que será siempre de supervisión. Estos dispositivos son una herramienta que se puede utilizar cuando el terapeuta considere, y no debe suponerle una inversión de tiempo excesiva que ralentice su trabajo. El objetivo principal de estos proyectos es que el niño se encuentre motivado por la terapia gracias al uso de un robot. Este proyecto fin de grado forma parte del proyecto NAOTherapist, el cuál deriva del proyecto Therapist y éste, a su vez, tiene como precursor el proyecto Ursus.

### 1.1.2.- Reconocimiento de poses

Desde tiempos ancestrales los seres humanos han intentado conocer el cuerpo humano, cómo funciona, las partes que lo componen, cómo curarlo y mejorarlo. En la búsqueda de conocimiento, el ser humano se interesó en el reconocimiento del movimiento para solucionar problemas que sufrían algunos individuos. El método médico más básico consistía en comparar el desempeño normal de una persona sana con el de la persona enferma. Analizando el funcionamiento del cuerpo en personas sanas se podían determinar indicadores que ayudasen a diagnosticar y curar la enfermedad a tratar.

En la antigua Grecia, a este estudio se le llamaba Quinesiología ('kinesis' - movimiento, 'logos' - estudio) y consistía en evaluar diferentes comportamientos y posturas de los individuos para poder determinar el estado normal del ser humano.

En la actualidad, gracias a la mejora de los sensores de captación de movimiento, los trabajos relacionados en esta vía de investigación están siendo muy impulsados en los últimos años. Se ha demostrado a través de numerosos trabajos que el lenguaje corporal de una persona muestra numerosas características de ésta de forma involuntaria y sin que el individuo sea consciente de ello. Un estudio reciente determina cómo las personas tienden a sincronizar sus movimientos con las demás personas con las que están interactuando, así como las personas varían los movimientos que realizan dependiendo de su estado de ánimo (Won *et al*, 2014).

Una de las vías dentro de la rama de investigación del reconocimiento de movimiento que más futuro tiene es aquella que se centra en la interacción humano-máquina. La robótica está en auge y que ésta pueda reconocer de forma precisa los movimientos de las personas sería fundamental para la mejora de los robots asistenciales (Fong & Nourbakhsh, 2013).

Con el estudio de los movimientos que realiza el cuerpo humano consciente o inconscientemente, se pueden determinar patrones o pautas de comportamiento que sigue el ser humano y que más adelante podrán utilizarse para que robots predigan qué movimiento va a realizar una persona y puedan esquivarla o pararlo según cuál sea la finalidad del robot o para corregir movimientos realizados de forma incorrecta o para identificar a personas por su forma de caminar, sentarse o movimiento de brazos, entre otros.

### **1.1.3.- Sistema Low-High**

Un sistema Low-High es una conversión de datos complejos de bajo nivel a unos más simples y entendibles por el ser humano en un nivel más alto de codificación. En este trabajo fin de grado se utiliza un sensor 3D el cual a través de sus sensores capta una imagen RGB <sup>1</sup> y una imagen de profundidad. Ambas imágenes están compuestas por una nube de puntos que se reciben a través de uno de los componentes que forman el proyecto y que éste envía para que el componente que los capte pueda procesarlos según sus necesidades.

En este trabajo, tal y como se hablará más adelante, los componentes desarrollados obtienen los datos que son enviados desde el componente que captura los datos del sensor 3D y los modelizan para poder identificar cada una de las partes del cuerpo humano. Estos datos procesados son utilizados más adelante por otros componentes que generalizan más los datos, como por ejemplo: “paciente sentado” o “paciente de pie”, o simplemente utilizan la información para comparar las poses de los pacientes.

Esta conversión de datos de bajo nivel a datos en alto nivel es necesaria para poder automatizar procesos de planificación, parte esencial del proyecto NAOTherapist para dotar de autonomía al robot terapeuta.

### **1.1.4.- Visión artificial**

La visión artificial es una parte de la inteligencia artificial encargada de la interpretación, reconstrucción y análisis de escenarios en 3D mediante imágenes recogidas a través de cámaras. En la literatura se han desarrollado una serie de modelos que asemejan la visión del ordenador a la visión humana y, para ello, ha sido necesario conocer cómo funciona el ojo humano en todos los campos de la ciencia, y así poder simular la complejidad de éste (Figura 6).

---

<sup>1</sup> RGB: Red, Green and Blue.

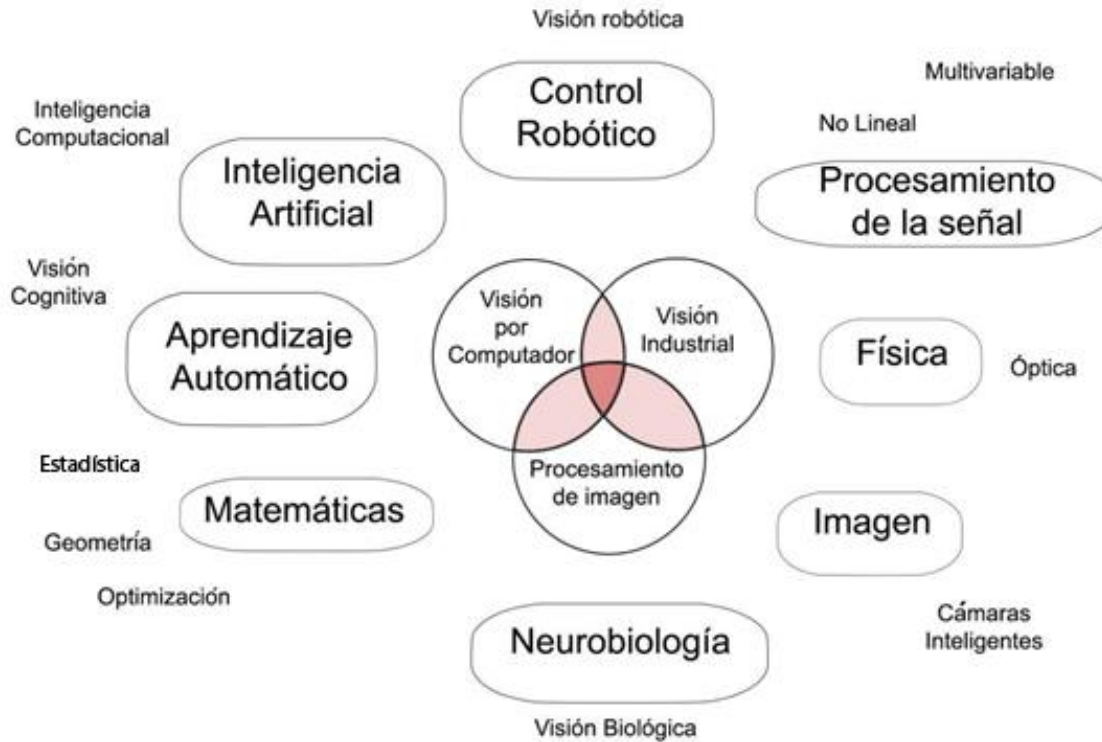


Figura 6. Campos relacionados con la visión artificial.

La visión artificial es uno de los campos en los que más se está investigando. Esto es debido a la importancia que tiene para el ser humano que un ordenador pueda ver y analizar su entorno y de esa forma poder tomar decisiones conforme a lo que ve.

Actualmente la visión artificial tiene numerosas aplicaciones, tales como:

- Reconocimiento de imagen: Un ejemplo podría ser la aplicación Goggles de Google, la cual, al obtener una imagen con nuestra cámara de fotos, el programa te devuelve todos los datos relacionados con dicha imagen.
- Reconocimiento de caracteres: Existen multitud de programas de OCR<sup>2</sup> que se dedican al reconocimiento de caracteres escritos.

Además de las aplicaciones anteriores, la visión artificial se emplea en una gran cantidad de campos, desde el reconocimiento de movimientos con sensor 3D Kinect<sup>3</sup> de Microsoft hasta la conducción autónoma o el reconocimiento de cáncer en una resonancia.

La visión del ordenador está compuesta por dos partes fundamentales: hardware y software. La parte hardware son las cámaras que captan las imágenes, mientras que la software está compuesta por

<sup>2</sup> OCR: Optical character recognition

<sup>3</sup> <https://www.microsoft.com/en-us/kinectforwindows/>



todos los programas que analizan y extraen información de estas imágenes para que éstas reconstruyan las imágenes para ser usadas más adelante.

## 1.2.- Objetivos

A continuación se detallarán los objetivos que se persiguen con este trabajo fin de grado que se justificarán a lo largo de este documento.

El objetivo principal es proporcionar al proyecto NAOTherapist de un módulo de visión y otro de aprendizaje. Para ello se tiene que tener en cuenta cuales son los objetivos del proyecto NAOTherapist y cómo funciona éste para adaptar este proyecto a sus necesidades a la vez que éste pueda ser generalizado y usar sus componentes en otros proyectos. Los objetivos del proyecto NAOTherapist son obtener un robot que realice planificaciones complejas de sesiones de rehabilitación, sea totalmente autónomo y realice una evaluación de la sesión y corrija los fallos del paciente en tiempo real. Para ello, este trabajo fin de grado se hace cargo de ciertos componentes de la arquitectura NAOTherapist, algunos de los cuales deben ser mejorados y otros diseñados desde cero, por lo que tiene como objetivos:

- Crear un nuevo modelo de ángulos del cuerpo humano que capture los datos del componente del sensor 3D y codifique los datos para que éstos puedan ser usados para la comparación de poses. Esto es necesario debido a la gran imprecisión de los datos recibidos por el componente del sensor 3D, por lo que se decide crear un modelo que sea más completo y preciso.
- Crear un componente de visión que obtenga los datos del componente que capta los datos del sensor 3D y genere una imagen en 3D con los datos que recibe, y sirva los datos a los componentes que se lo soliciten.
- Crear un componente de aprendizaje de nuevas poses que se conecte con el componente de visión, reciba información del componente de Planificador de Terapias y guarde las nuevas poses propuestas por el terapeuta.
- Crear una interfaz gráfica que ayude al terapeuta a grabar nuevas poses y ejercicios, en la cual se introducirán las características del ejercicio a grabar, y se mostrarán las poses que componen cada ejercicio. Ésta debe poder mostrar la pose y los ejercicios que se van a almacenar.

## 2.- Estado de la cuestión

A continuación se expondrá el estado actual de las investigaciones sobre el tema que trata el proyecto fin de grado, la tecnología referente al sensor 3D utilizado para realizar este trabajo y el estado del proyecto NAOTherapist del cual forma parte este trabajo.

### 2.1.- NAOTherapist

NAOTherapist es un proyecto que forma parte de Therapist (Calderita, 2013) y se encuentra en una fase muy temprana de desarrollo. Al igual que Therapist, está basado en una arquitectura por componentes, por lo que todas sus funciones son independientes y pueden ser sustituidas por otras, lo que proporciona una gran capacidad de escalabilidad, de ampliación de funcionalidades y de intercambio de componentes entre diferentes proyectos. NAOTherapist no es una arquitectura ad-hoc basada en planificación ya que puede utilizar cualquier tipo de robot realizando un componente específico para cada uno de ellos que realice un retargeting de los datos proporcionados por el componente de visión desarrollado en este documento.

En el proyecto NAOTherapist se utiliza la modularización de Robocomp (Manso *et al*, 2010) que proporciona un entorno de desarrollo, herramientas y reutilización de componentes destinados al control de plataformas robóticas. La comunicación entre componentes está realizada con interfaces Ice, las cuales proporcionan una comunicación entre componentes independiente del lenguaje en el que este escrito cada módulo.

El proyecto NAOTherapist está formado por los componentes expuestos en la arquitectura mostrada en la Figura 8. Aquí sólo serán explicados los módulos que no pertenecen a este trabajo, ya que el resto se tratarán más adelante en este documento.

- Interfaz de Usuario: Es una interfaz donde el médico rehabilitador y/o el terapeuta introducen los problemas del paciente para que el módulo de generación del problema pueda generar la sesión que deberá seguir el paciente (Figura 7).

Figura 7. Interfaz del médico para diseñar las terapias de NAOTherapist.

- Diseñador de terapias: Este módulo recibe la terapia introducida por el médico rehabilitador y planifica los ejercicios (nivel alto de planificación), busca en la base de datos si tiene suficientes ejercicios para poder planificar una sesión de rehabilitación que se adecúe a los requisitos propuestos por el médico rehabilitador y/o terapeuta. En caso de no disponer de suficientes ejercicios para realizar una sesión, este módulo se lo comunicará al componente de aprendizaje, el cual será desarrollado en este trabajo, y que será el encargado de grabar las nuevas poses y ejercicios que serán propuestos por el médico rehabilitador y/o terapeuta en la base de conocimiento para que el diseñador de terapias pueda realizar una sesión para el paciente.

Una vez se ha generado la sesión, se genera una salida en formato PDDL para que el componente Decision Support lo reciba.

- Decision Support: Este componente pertenece al framework RoboComp y es el encargado de la planificación de medio nivel. Esto quiere decir que es el componente encargado de que la sesión planificada se lleve a cabo correctamente.

Este componente se encarga de enviar las acciones, a medida que analiza que el paciente está realizando las acciones correctamente. Las acciones que envía este componente pueden ser para corregir al paciente, motivarlo o para obtener la pose siguiente, entre otras muchas.

- Executive: Este módulo es el encargado de traducir la acción planificada por el componente Decision Support a un lenguaje que entienda el robot. En el caso del proyecto NAOTherapist, el mó-

dulo ejecutivo será el encargado de traducir la acción para el que el robot NAO comprenda la acción que debe ejecutar.

- NAO Robot: Este componente es el encargado de realizar el retargeting específico del robot NAO. Al tratarse de un proyecto por componentes, este componente sería el único que habría que cambiar en caso de que se cambiase el robot NAO por otro.

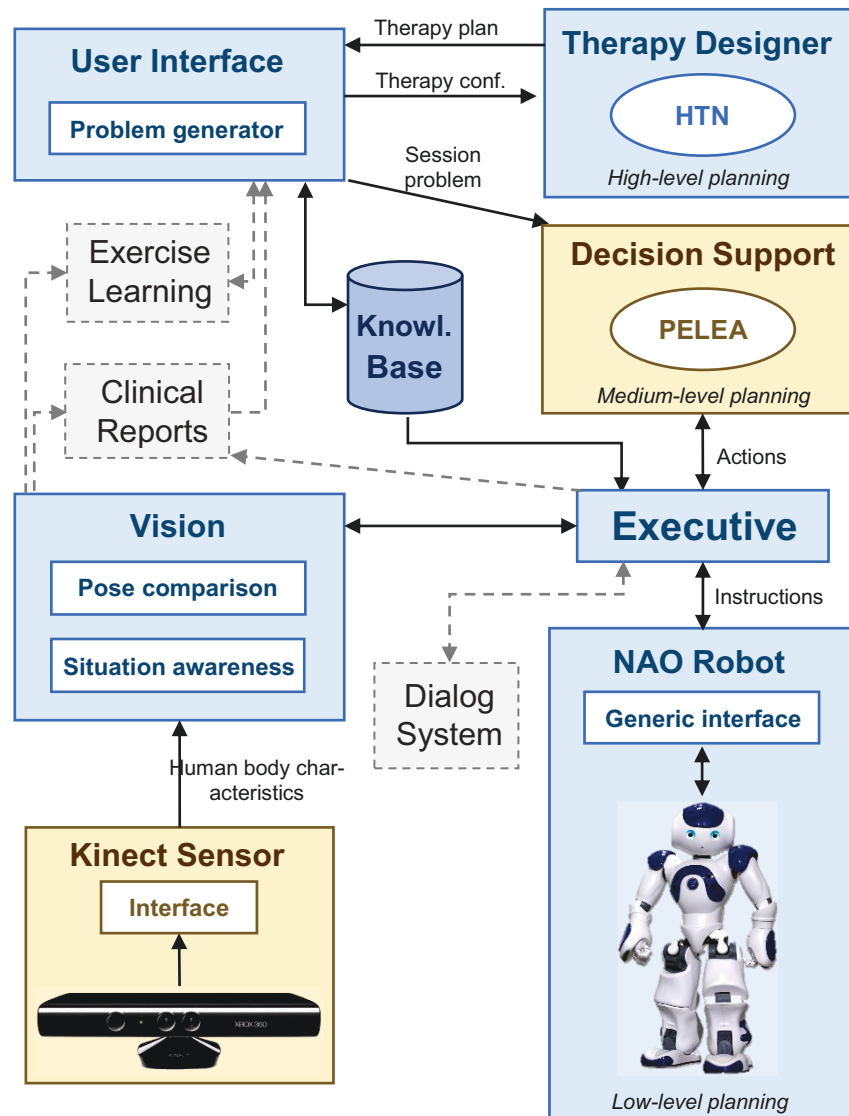


Figura 8. Diagrama de la arquitectura de NAOTherapist.

Este proyecto está compuesto por tres niveles de planificación claramente definidos:

- Nivel alto de planificación: Se añaden las poses y los ejercicios a la base de conocimiento para un posterior uso.
- Nivel medio de planificación: Las poses y ejercicios almacenados en la base de conocimiento son utilizados para planificar las sesiones de rehabilitación.

- Nivel bajo de planificación: Las poses utilizadas para la sesión de rehabilitación son enviadas al robot para que éste las utilice durante la sesión.

Las poses que han sido recogidas desde el sensor 3D Kinect, son utilizadas para realizar una sesión de rehabilitación con el paciente y enviadas al robot para que éste las interprete.

## 2.2.- Historia

A continuación se desarrollarán en profundidad los trabajos que están más relacionados con el proyecto en general y se mostrará su funcionamiento y las diferencias con el proyecto del que trata este documento y más adelante se explicarán brevemente otros proyectos de los cuales se han obtenido ideas y que han influenciado al realizar este proyecto fin de grado.

YouMove es un trabajo que trata sobre un programa que permite a los usuarios grabar poses y sesiones capturando imágenes a través del sensor Kinect. Está orientado al ballet aunque puede ser usado con otros fines (Fraser *et al*, 2013). Para la implementación se dividió en dos programas:

- Software de grabación: Se implementó en el lenguaje C#, las herramientas utilizadas fueron el WPF<sup>4</sup> framework y Microsoft Kinect SDK<sup>5</sup> y su interface es como se muestra en la Figura 9.

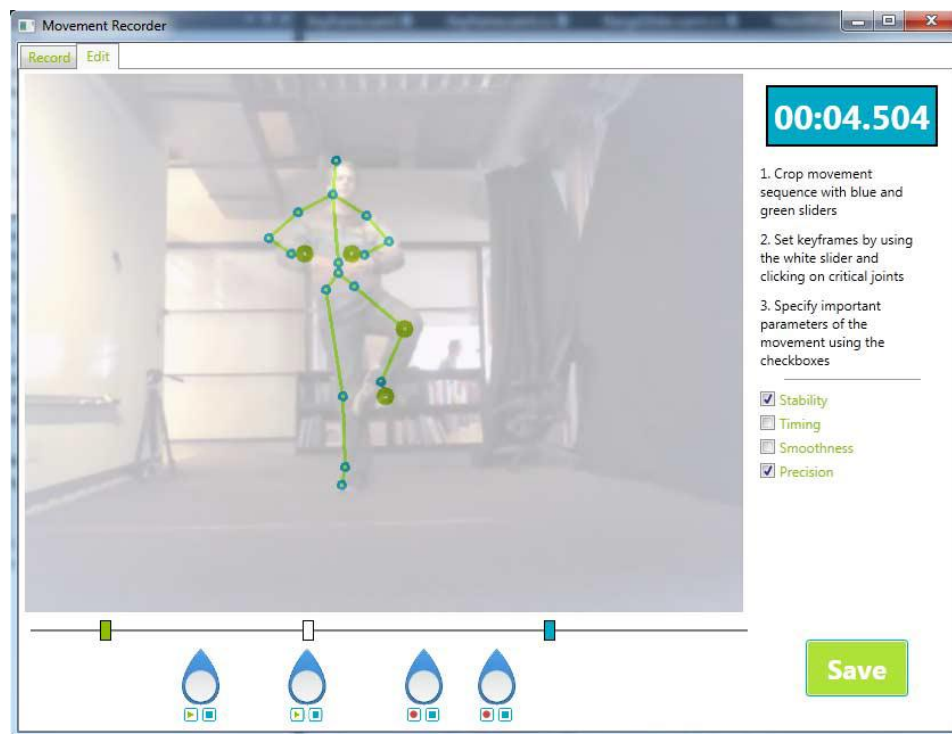


Figura 9. Interfaz de grabación de YouMove.

<sup>4</sup> WPF: Windows Presentation Foundation

<sup>5</sup> SDK: software development kit

A través de la interfaz gráfica se puede ver una imagen del usuario y la reconstrucción del esqueleto sobreimpresa, así como las poses que se han grabado. También se pueden organizar las poses durante la sesión para modificar la posición de cada una. Cuando se presiona el botón “Save”, se genera un archivo de video con la sesión grabada así como un archivo de texto plano con la localización de cada punto del esqueleto en los tiempos indicados.

En cambio, en el proyecto del que trata este documento, no se obtiene la imagen del individuo directamente, únicamente se realiza una reconstrucción del esqueleto. Esto es debido a que YouMove utiliza directamente el SDK de la Kinect, mientras que este proyecto utiliza el componente WinKinect-Comp, que se explicará más adelante, y que es uno de los requisitos que tiene el proyecto NAOTherapist.

- Software de entrenamiento: Se implementó en el lenguaje C++, las herramientas utilizadas fueron openFrameworks, OpenGL y Microsoft Kinect SDK. Se dispone de un menú (Figura 10).



Figura 10. Menú de entrenamiento de YouMove.

Se muestran las opciones que tiene el software de entrenamiento. La guía de postura se muestra como en la Figura 11, la cual indica la posición a mantener y el tiempo que se tiene que mantener, a la vez que indica los errores que tiene la postura tomada.

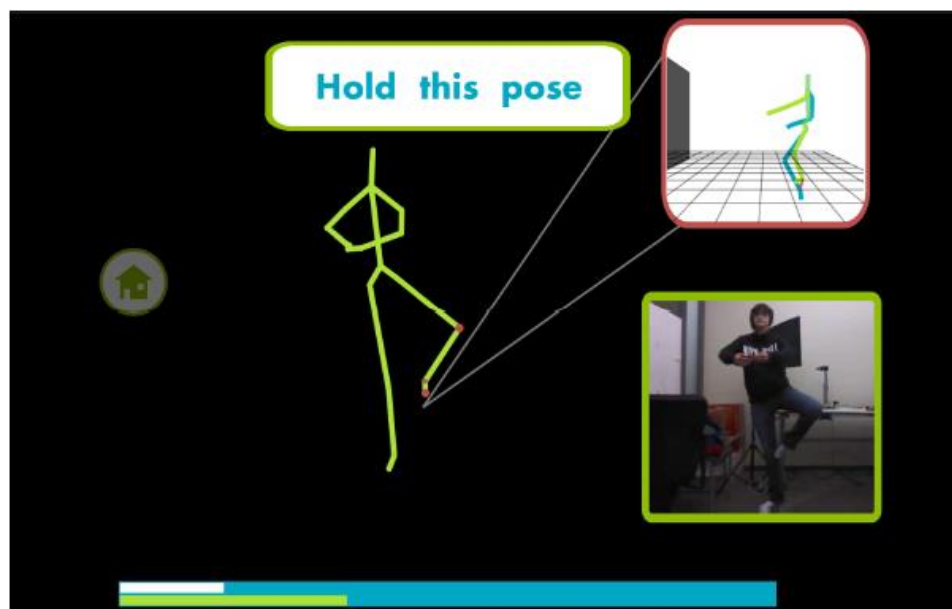


Figura 11. Grabación de poses con YouMove.

Cuando finalizas la guía muestra una puntuación y lo compara con la postura correcta (Figura 12).

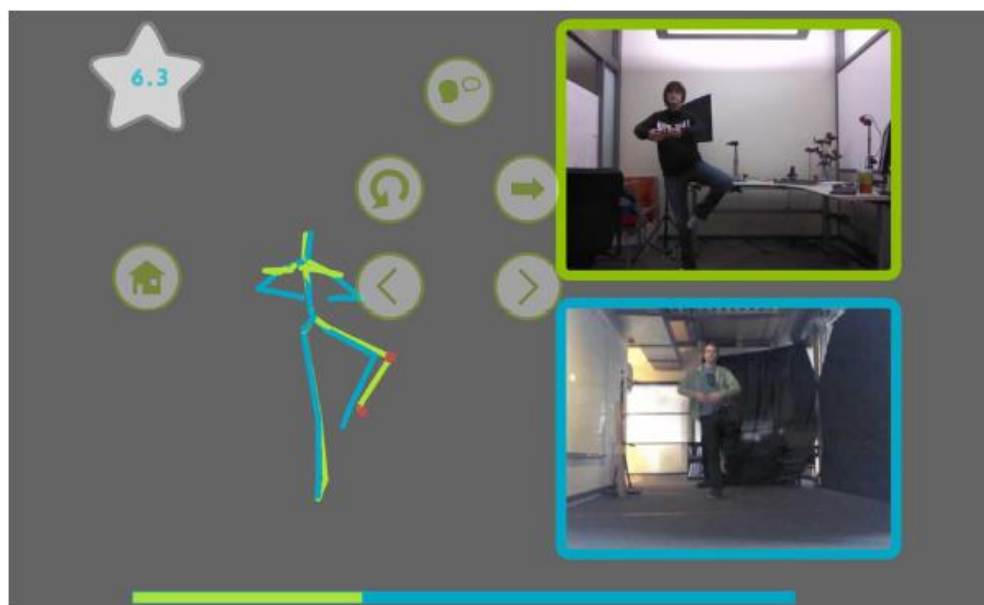


Figura 12. Comparación de poses con YouMove.

En este documento se detalla, más adelante, la grabación de poses, su comparación con los datos de la base de conocimiento, pero no realiza una comparación gráfica de las mismas, únicamente se realiza una comparación para no almacenar más de una vez una misma postura, ya que la finalidad del proyecto es la grabación de nuevas poses. La fase de analizar la pose correcta respecto a la que debe realizar el usuario, estaría a cargo de otro componente del proyecto NAOTherapist.

El modelo del esqueleto que se ha desarrollado en el siguiente trabajo tiene en cuenta ciertos ángulos y criterios para poder hacer retargeting entre el esqueleto de la Kinect y un robot humanoide (Alfaro, 2012). En este trabajo el autor utiliza el sensor 3D Kinect de Microsoft para realizar la teleoperación con el robot Nao, todo ello programado en lenguaje ROS. Se compone de varias capas (Figura 13).

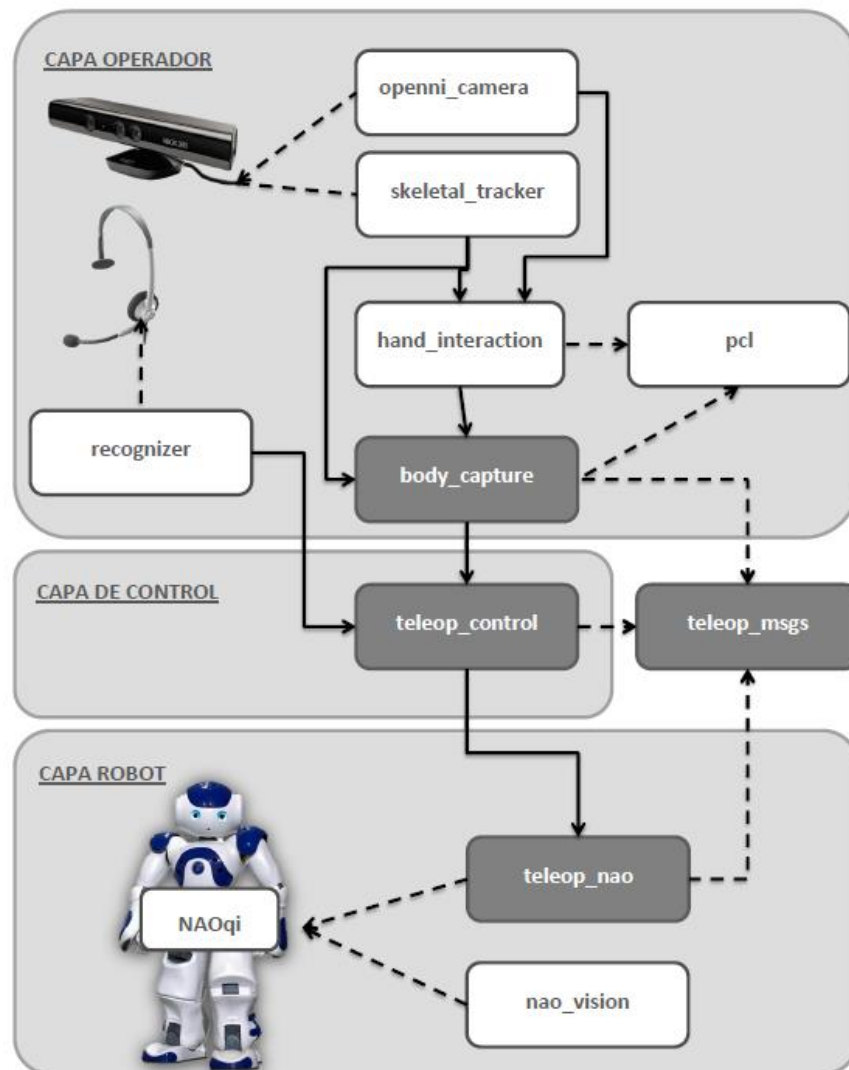


Figura 13. Diagrama de la arquitectura de teleoperación (Alfaro, 2012).

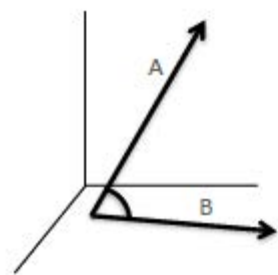
- Capa operador: Es la encargada de capturar la información de la Kinect. Utiliza la API OpenNI<sup>6</sup> para la recopilación de los datos de la Kinect y su envío a través de ROS. En este proyecto, esta función la realizaría el componente WinKinectComp.

<sup>6</sup> <http://www.openni.org>



- Capa de control: Es la encargada de procesar los datos recogidos por la capa del operador y transformarlos para que puedan ser usados por la capa del robot. En el proyecto, esta función la realizará el componente VisionComp.
- Capa robot: Es la encargada de interactuar con el robot Nao, mediante la interfaz NAOqi. En este proyecto esta fase la realiza el componente ExecutiveComp.

El componente VisionComp reconstruye un esqueleto y lo muestra en una ventana en forma de video, función que esta capa de control no realizaría, y el cálculo de ángulos realizado por esta capa sería el siguiente: los datos obtenidos de la Kinect se calculan los ángulos de la mayoría de las articulaciones, usando para ello, la fórmula de la Figura 14.



$$\alpha = \arccos \frac{A \cdot B}{|A||B|}$$

Siendo:

$$A \cdot B = A_x B_x + A_y B_y + A_z B_z$$

$$|A| = \sqrt{A_x^2 + A_y^2 + A_z^2}$$

$$|B| = \sqrt{B_x^2 + B_y^2 + B_z^2}$$

Figura 14. Cálculo de ángulos (Alfaro, 2012).

La amplitud del hombro la calcula según se muestra en la Figura 15.

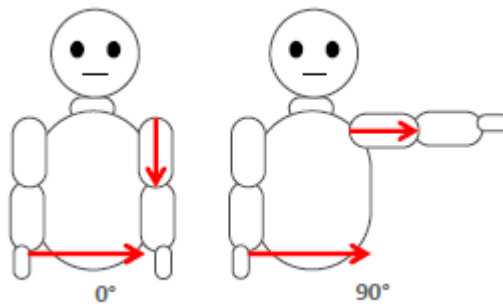


Figura 15. Amplitud del hombro (Alfaro, 2012).

Así como la rotación del hombro sería como la Figura 16.

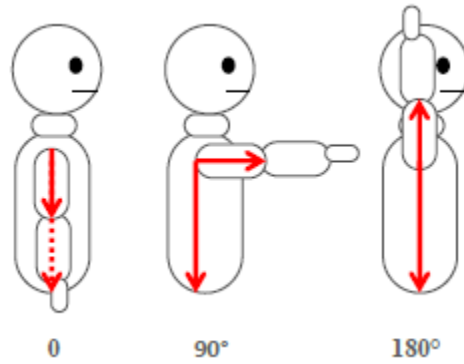


Figura 16. Rotación del hombro (Alfaro, 2012).

Y la amplitud del codo, se calcularía como muestra Figura 17.

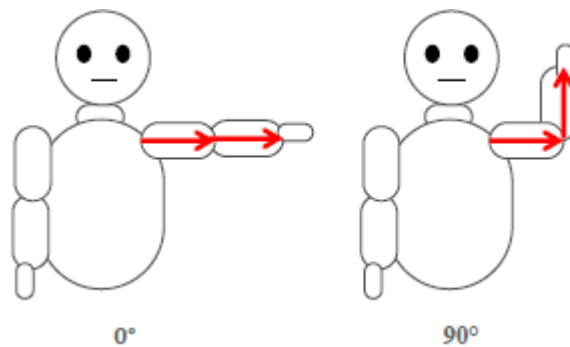


Figura 17. Amplitud del codo (Alfaro, 2012).

La única articulación que se calcularía de forma diferente al resto sería la rotación del codo cuyo cálculo depende de la posición del codo, la cuál podría ser como se muestra en la Figura 18 o en la Figura 19.

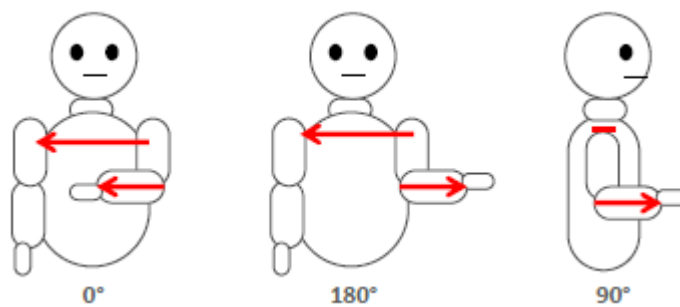


Figura 18. Codo cerrado (Alfaro, 2012).

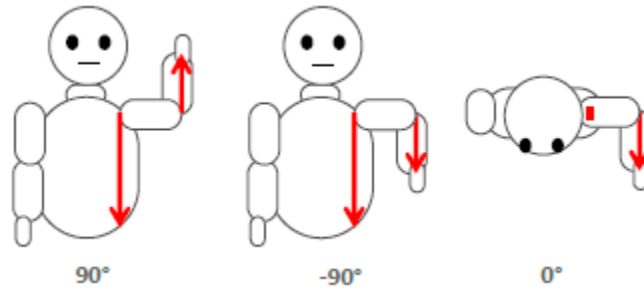


Figura 19. Codo abierto (Alfaro, 2012).

Para realizar el cálculo de esta articulación el autor creó una fórmula para hallar de la forma más precisa posible el dato necesario, tal y como muestra la Figura 20.

$$RC = HC \times \frac{AH}{90^\circ} + HH \times \left(1 - \frac{AH}{90^\circ}\right)$$

- $RC \equiv$  Ángulo de rotación del codo.
- $HC \equiv$  Ángulo de rotación del codo calculado mediante el vector que une el hombro a la cadera.
- $HH \equiv$  Ángulo de rotación del codo calculado mediante el vector que une los hombros.
- $AH \equiv$  Ángulo de apertura del hombro.

Figura 20. Fórmula de la rotación del codo (Alfaro, 2012).

Usando la fórmula de la Figura 20, el cálculo de la articulación se realizaría como en el ejemplo de la Figura 21.

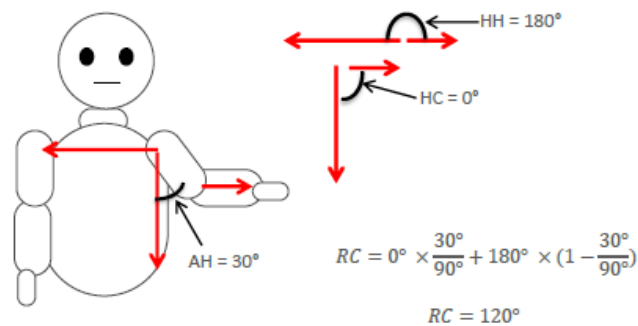


Figura 21. Ejemplo de rotación del codo (Alfaro, 2012).

A lo largo de este documento se detallará cómo el cálculo de ángulos realizado por el componente VisionComp se realiza a través de planos respecto al individuo con lo que se calculan los ángulos de las articulaciones. De ésta forma si el individuo no está completamente perpendicular al dispositivo Kinect el cálculo de los ángulos no se ve afectado debido a la robustez del modelo antropomórfico que se ha dise-

ñado y produce un cálculo más genérico, preciso y una traducción más directa a los sistemas robóticos, debido a la utilización de este sistema de planos. El cálculo de la rotación del codo se ha realizado utilizando el cálculo formulado en el trabajo de Santiago Alfaro, ya que esta medida ha probado a ser la más efectiva.

Otros trabajos que se han tenido en cuenta a la hora de realizar este proyecto han sido los relacionados con la clusterización del problema y su análisis en tiempo real para el reconocimiento de poses (O'Hara *et al*, 2012).

El proyecto del equipo de Stephan O'Hara realiza un sistema no supervisado de reconocimiento de acciones de un video en tiempo real que partiendo de una base sin conocimiento previo, crea clústeres de las acciones usando la distancia del "Product Manifold" (Lui *et al*, 2010) y realizan la detección de las poses nuevas a través de los ejemplos aprendidos anteriormente (Figura 22).

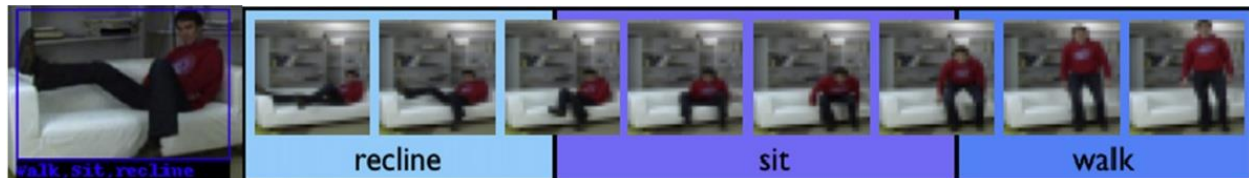


Figura 22. Reconocimiento de pose (O'Hara, 2012).

Sin embargo, su función es la de reconocer poses utilizando para ello sistemas de clusterización. En este trabajo fin de grado no se contempla el aprendizaje no supervisado, ya que hay que comparar las poses de nuestra base del conocimiento con la pose actual del paciente y añadirla a nuestra base de conocimiento en el caso de que esa pose sea nueva, todo ello en tiempo real.

El sensor 3D Kinect y el reconocimiento del lenguaje para comunicarse con el robot para que éste adquiriera conocimientos nuevos en tiempo real, son dos herramientas que son muy utilizadas en la investigación debido a su potencial. González-Pacheco (Gonzalez-Pacheco *et al*, 2013), logró que un robot reconociera una postura que adquiriría una persona, pero era la persona la que le decía, mediante comandos de voz, como quería que etiquetase esa postura y el robot la añadía a su base de conocimiento (Figura 23).

Este sistema necesita de la interacción de un usuario para clasificar las posturas y no disponía de ninguna autonomía ya que siempre dependería de un usuario que le ayudara para disponer de una funcionalidad completa. Este sistema es similar al introducido en este trabajo fin de grado, ya que durante la captura de poses por parte del componente LearningComp, realiza una comparación entre la postura del paciente y las posturas de la base de conocimiento para analizar si dicha postura es nueva o ya conocida.

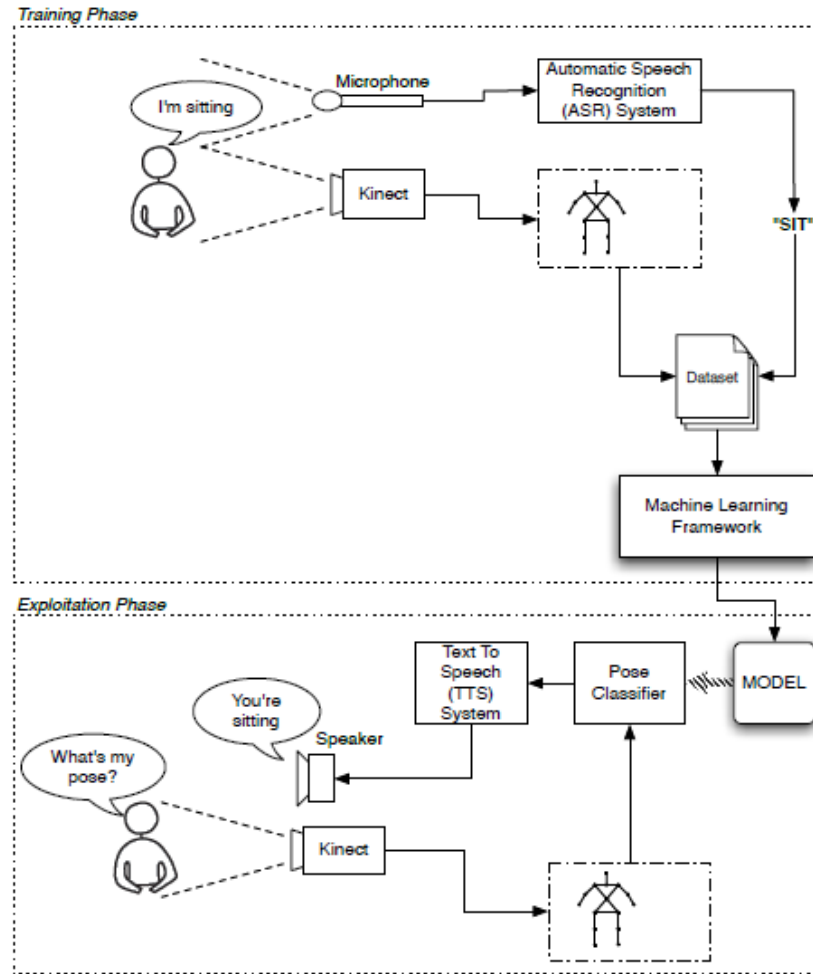


Figura 23. Diagrama del modelo (González-Pacheco et al, 2013).

En el estudio realizado por Bednarik y Herman (Bednarik & Herman, 2015), los autores hacen uso de un dispositivo Kinect para detectar la posición y las posturas del sujeto a estudiar (Figura 24). A diferencia del trabajo desarrollado en este documento, para el estudio de las posturas Bednarik y Herman hacen uso de un autómata temporal conforme al estado del cuerpo, la movilidad de las articulaciones y las limitaciones de tiempo, mientras que en este documento se tendrán en cuenta los ángulos formados por las articulaciones.



Figura 24. Detección de pose (Bednark & Herman, 2015).

## 2.3.- Kinect

El sensor Kinect fue creado por Alex Kipman para la empresa Microsoft y formaba parte del proyecto Natal. La empresa Microsoft la comercializó como accesorio para su videoconsola Xbox 360 aunque más adelante se vio todas las posibilidades que ofrece este aparato y se ofreció un SDK por parte de Microsoft para poder conectar y manejar la Kinect desde un ordenador.

Este sensor ofrece la posibilidad de interactuar con el sistema con el cuerpo humano, gracias a su cámara en 3D, lo que permite que su curva de aprendizaje sea casi nula, ya que al reconocer el movimiento, el usuario no tiene que aprender a utilizar el dispositivo, sino comportarse de la forma más natural posible (Figura 25).



*Figura 25. Jugadores de la Xbox con Kinect.*

La tecnología utilizada por la Kinect fue creada por PrimeSense, una empresa israelita especializada en las interfaces naturales de usuario, y el software lo creó la empresa Rare, perteneciente a Microsoft.

Aparte de sus especificaciones técnicas donde realmente destaca este sensor respecto a la competencia es en el precio y en el software. La Kinect fue el primer sensor 3D que ofrece poder detectar hasta 6 personas a la vez y diferenciar esas personas del entorno que tienen a su alrededor, y a su vez, es capaz de generar el esqueleto de la persona con una tasa de 30fps<sup>7</sup>. Si a estas funcionalidades le unimos el bajo

---

<sup>7</sup> Fps: Frames per second.

coste que supone adquirirlo, se ha convertido en el sensor 3D más extendido a nivel de investigación y desarrollo en el ámbito académico.

La Kinect está formada por los siguientes componentes como muestra la Figura 28:

- Cámara RGB: Obtiene una imagen en color. Es una cámara VGA<sup>8</sup>, con una resolución de 1080p, con una tasa de 30 fps y 32 bits de color.
- Cámara de profundidad: Obtiene una imagen de profundidad. analizando los datos obtenidos por el sensor infrarrojo (Figura 26). Es un sensor CMOS<sup>9</sup> monocromo que captura los datos emitidos por el sensor infrarrojo. Con una resolución de 320x240, con una tasa de 30 fps y 16 bits de profundidad (Figura 27).

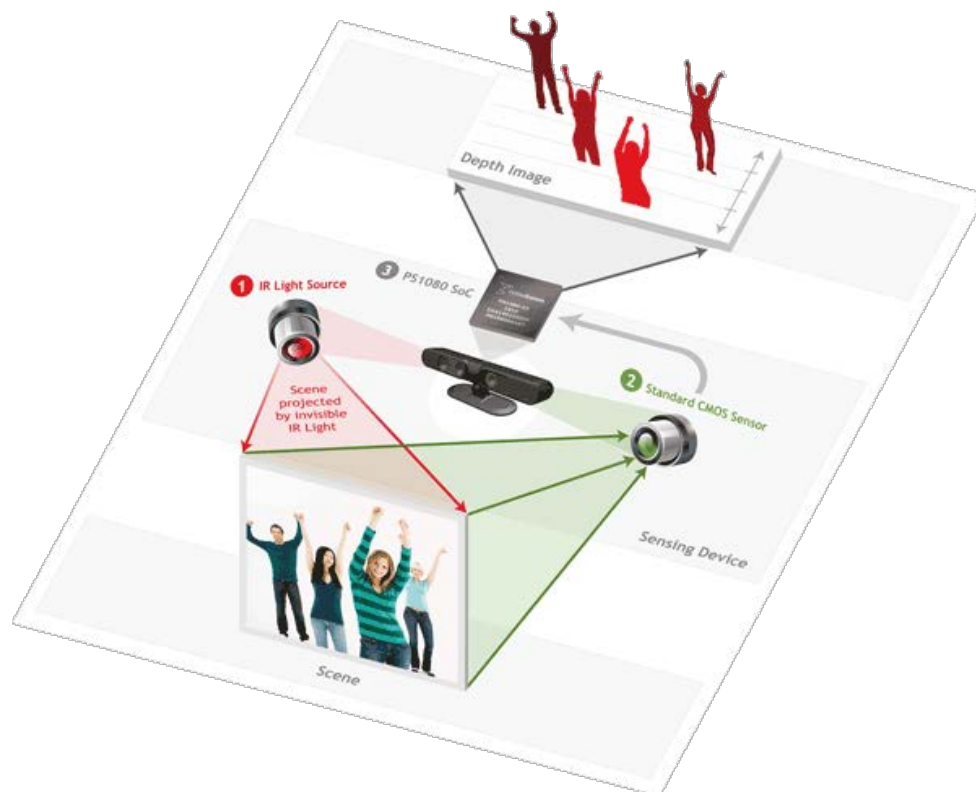


Figura 26. Cámara de profundidad de la Kinect.

<sup>8</sup> VGA: Video Graphics Array

<sup>9</sup> CMOS: Complementary metal-oxide-semiconductor

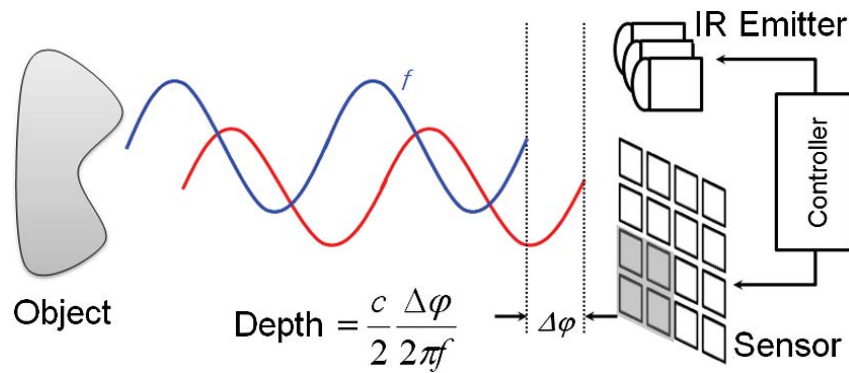


Figura 27. Funcionamiento de la cámara de profundidad de la Kinect.

- Sensor infrarrojo: Sirve para que la cámara de profundidad obtenga la imagen, proyectando un haz de infrarrojos por la habitación (Figura 27).
- Micrófono multiarray: Graba el audio. Consiste en la colocación de múltiples micrófonos a lo largo del dispositivo para poder determinar de dónde procede el sonido.

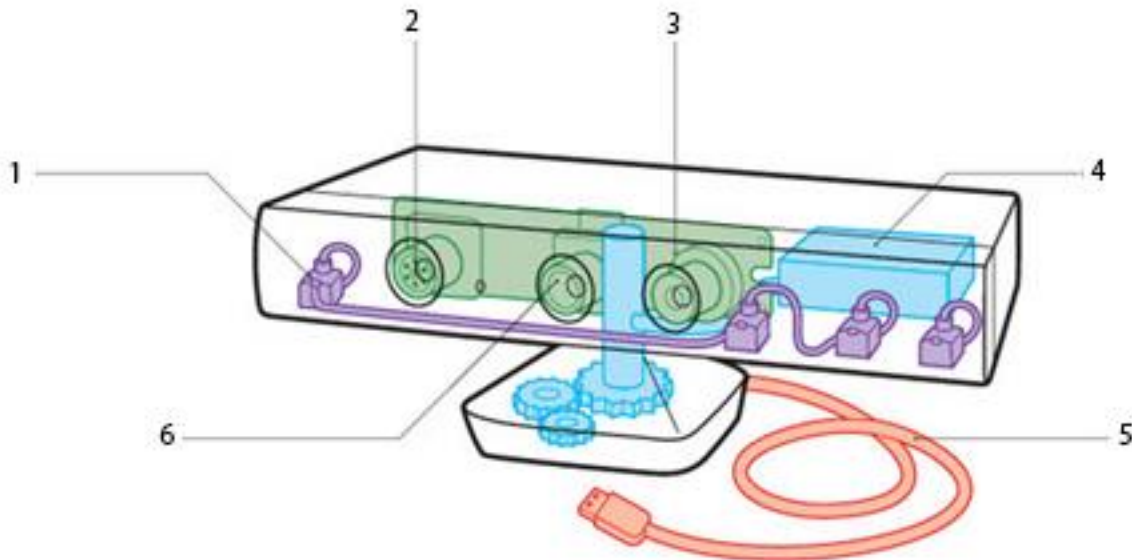


Figura 28. Componentes de la Kinect..

(1) Micrófonos. (2) Sensor IR. (3) Cámara Profundidad. (4) Motor. (5) USB. (6) Cámara RGB.

La distancia recomendada por Microsoft para el uso de este dispositivo es de 1.2-2.5m, aunque en un rango de 0.5-3.5m los resultados son aceptables. Este dispositivo tiene un ángulo de visión de 57° en horizontal y 43° en vertical, además su base permite un movimiento de 27° arriba y abajo.

La Kinect al capturar el cuerpo humano a través de sus sensores obtiene los puntos del esqueleto (Figura 29).



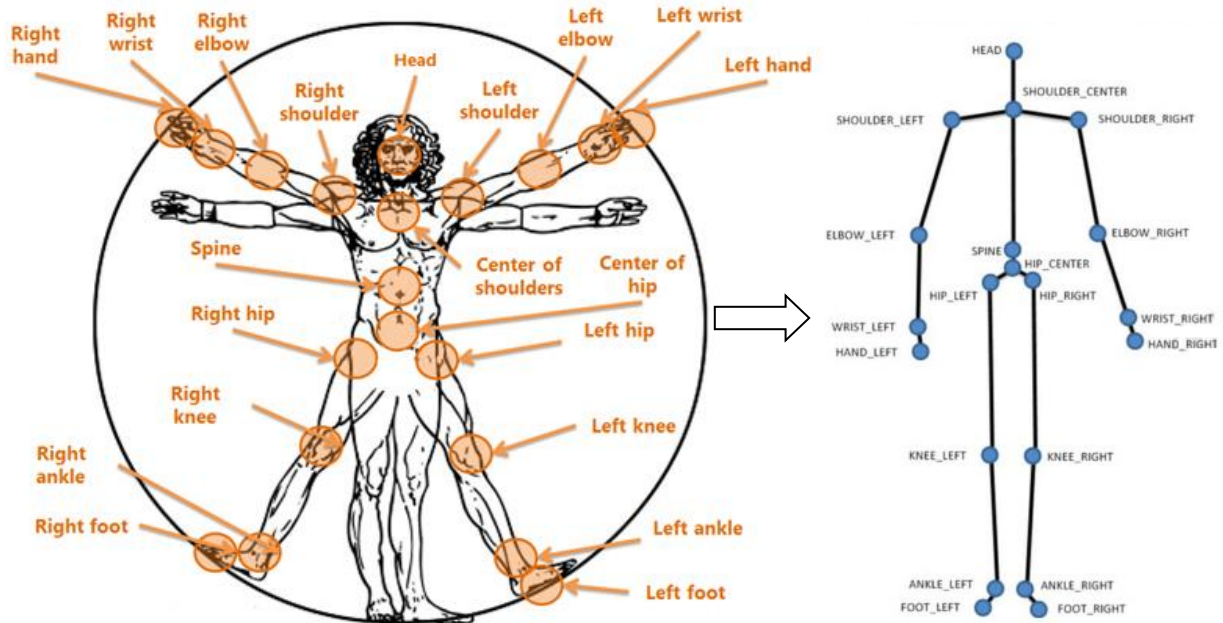


Figura 29. Articulaciones del cuerpo humano y el esqueleto para la Kinect.

Como puede observarse en la Figura 30, con el sensor de profundidad obtenemos una imagen en blanco y negro de lo que hay delante del sensor, mientras que la cámara RGB obtiene la imagen en color.

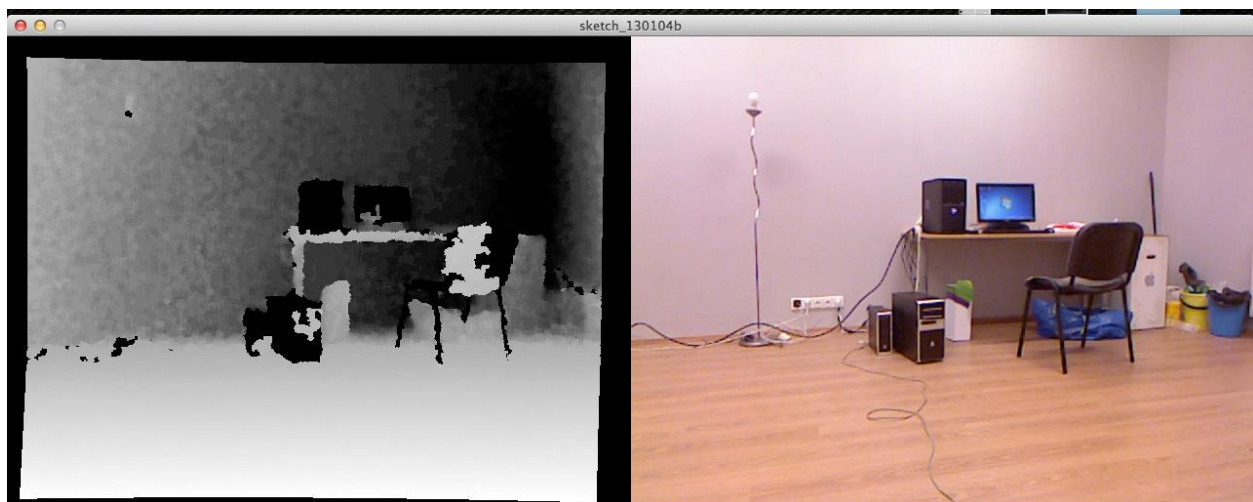
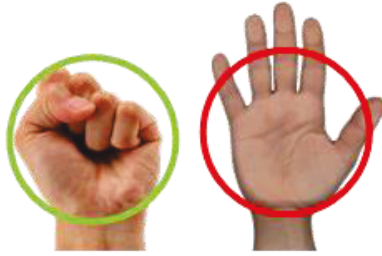


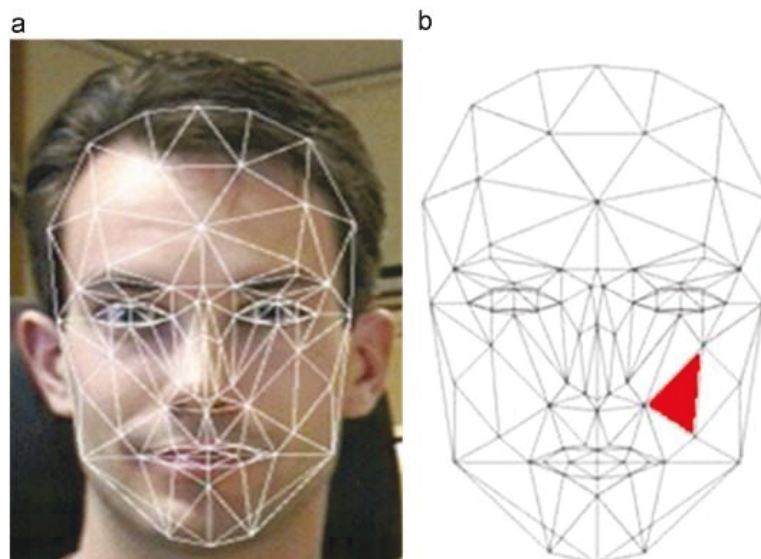
Figura 30. Imagen RGB y Profundidad de la Kinect.

La Kinect a través del software de Microsoft es capaz de detectar si una mano está abierta o cerrada (sin indicar el grado de apertura) como muestra la Figura 31, por lo que en este apartado la Kinect es algo imprecisa. Existen numerosos trabajos en los cuales si se consigue identificar perfectamente la mano y los dedos. (Raheja *et al*, 2011).



*Figura 31. Reconocimiento de la mano.*

Otra de las funciones que hacen del sensor Kinect un dispositivo novedoso y muy usado en la actualidad es su reconocimiento de caras. Tal y como se ve en la Figura 32, la Kinect traza un mapa de triángulos de la cara y posteriormente lo analiza para poder identificar la posición de los ojos, boca o nariz, entre otros muchos datos que devuelve. Existen estudios que mejoran este procesamiento de los datos para ser aún más preciso (González, 2014) o (Li *et al*, 2013).



*Figura 32. Reconocimiento de la cara.*

## 2.4.- RoboComp<sup>10</sup>

Es un framework Open-Source creado por el laboratorio ROBOLAB<sup>11</sup> de la Universidad de Extremadura. El framework está implementado en su mayoría en C++<sup>12</sup> y Python<sup>13</sup>, aunque el componente de interacción directa con el sensor Kinect está implementado en C#<sup>14</sup> a través de Visual Studio y todos los componentes se comunican entre sí utilizando el lenguaje Ice<sup>15</sup>. Es modulable, esto quiere decir que se pueden ampliar las funcionalidades del mismo añadiendo nuevos componentes independientes.

### 2.4.1.- ZeroICE

Ice es un lenguaje de programación orientado a objetos que se utiliza para la interacción a bajo nivel de las comunicaciones entre interfaces. Este lenguaje permite que se desarrolle el programa principal en Java, C++, Python, entre otros, y realizar la comunicación de forma independiente al lenguaje del programa principal.

En el proyecto NAOTherapist todos los componentes que lo forman se comunican mediante Ice ya que al ser independiente del lenguaje del componente, es más sencillo poder introducir nuevos componentes al proyecto.

### 2.4.2.- WinKinectComp

Es un componente que forma parte del framework Robocomp. Este componente está implementado en C# a través de Visual Studio, utiliza el Microsoft Kinect SDK, el cual permite a este componente poder interaccionar con la Kinect y poder pre procesar los datos para enviarlos según la modelación elegida. Este componente se encarga de obtener los datos de la Kinect, de bajo nivel, y enviarlos a través de una interface IceStorm, para que cualquier otro componente pueda usarlos.

IceStorm es un servicio de publicador-suscriptor, esto significa que el programa publicador emite los datos cuando se desencadena un evento, y todos los suscriptores que estén enlazados recibirán los datos simultáneamente (Figura 33).

---

<sup>10</sup> <http://robocomp.sourceforge.net/wordpress/>

<sup>11</sup> <https://robolab.unex.es/>

<sup>12</sup> <http://www.cplusplus.com/>

<sup>13</sup> <https://www.python.org/>

<sup>14</sup> <https://msdn.microsoft.com/es-es/library/kx37x362.aspx>

<sup>15</sup> <https://www.zeroc.com>

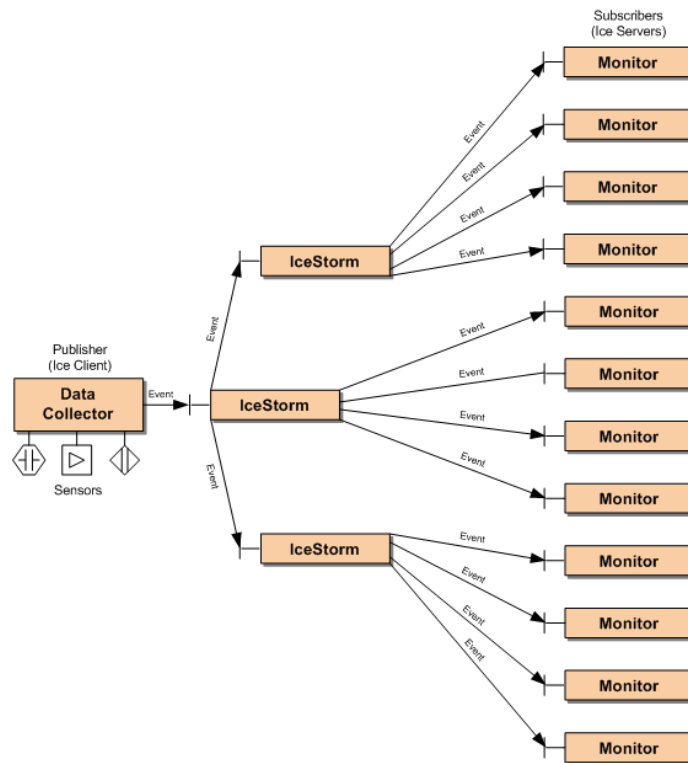


Figura 33. Diagrama de IceStorm.

El modelo detallado será adjuntado en el apartado 7.1.- Modelo de WinKinectComp por lo que en esta sección se realizará una visión más general de él.

El Módulo RoboCompMSKBody es el encargado de enviar toda la información correspondiente a los cuerpos de las personas que son detectadas por el sensor 3D Kinect. La clase TPerson es la encargada de proporcionar los datos de la persona detectada y está formada por: un identificador, un conjunto de articulaciones (JointsList), un estado de la persona, una posición geométrica y un conjunto de huesos (BoneOrientations) (Figura 34). La interfaz MSKBody sirve para rellenar la estructura de datos del módulo y es enviada por el componente WinKinectComp, y MSKBodyEvent es la interfaz que hay que implementar en el componente destino para recibir los datos que envía el componente WinKinectComp.

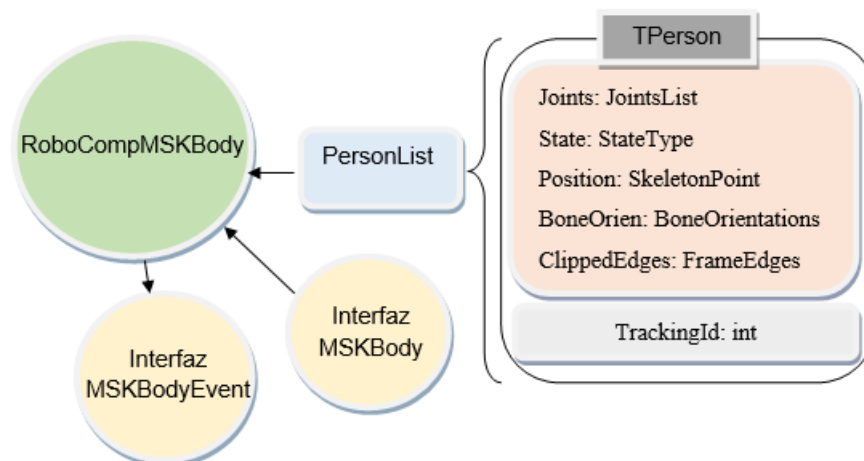


Figura 34. Módulo RoboCompMSKBody del modelo WinKinectComp.

El Módulo RoboCompMSKFace es el encargado de enviar toda la información correspondiente a la cara de las personas que son detectadas por el sensor 3D Kinect (Figura 35). DetailedFace es la clase que almacena toda la información referente a la cabeza de la persona reconocida por el sensor, como por ejemplo: la posición geométrica de la cabeza, la posición de los ojos, imagen de la cabeza, la división de la cara en diferentes triángulos para realizar una posterior verificación facial, un identificador o la inclinación de la cabeza respecto a los planos del cuerpo.

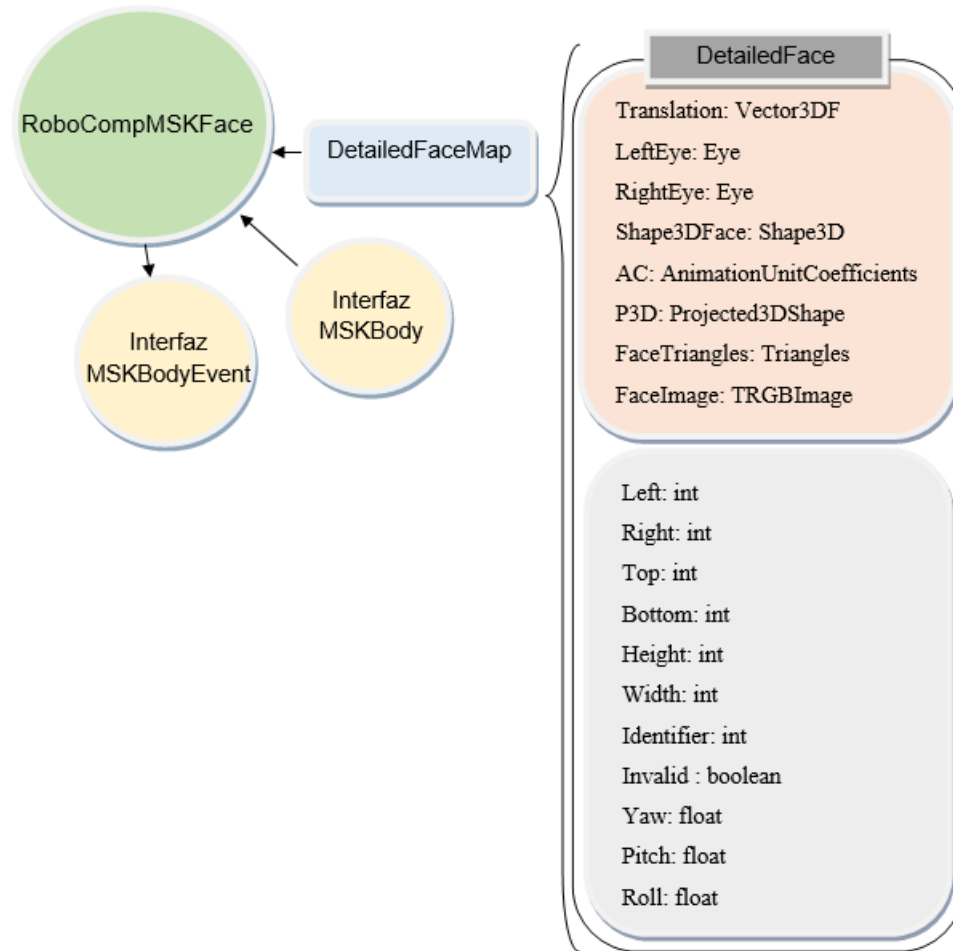


Figura 35. Módulo RoboCompMSKFace del modelo WinKinectComp.

El Módulo RoboCompMSKRGBD es el encargado de enviar toda la información correspondiente a las imágenes RGB y de profundidad detectadas por el sensor 3D Kinect (Figura 36). Está compuesto por la imagen en color (TRGBImage) y la imagen de profundidad (TdepthImage).

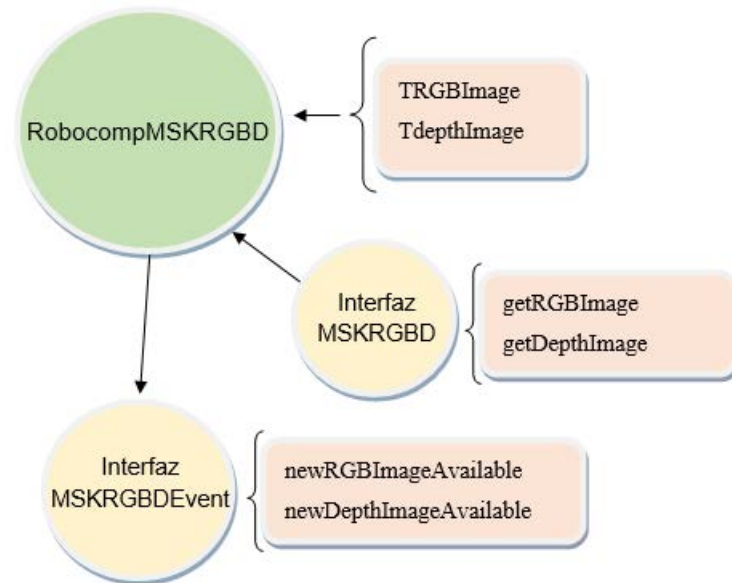


Figura 36. Módulo RoboCompMSKRGBD del modelo WinKinectComp.

El Módulo RoboCompMSKHands es el encargado de enviar toda la información correspondiente a la manos de las personas que son detectadas por el sensor 3D Kinect (Figura 37). UserInfo es una clase que identifica el esqueleto al que pertenecen esas manos (SkeletonTrackingId) y el conjunto de las manos (HandPointers) que estará formado por la mano izquierda y la mano derecha.

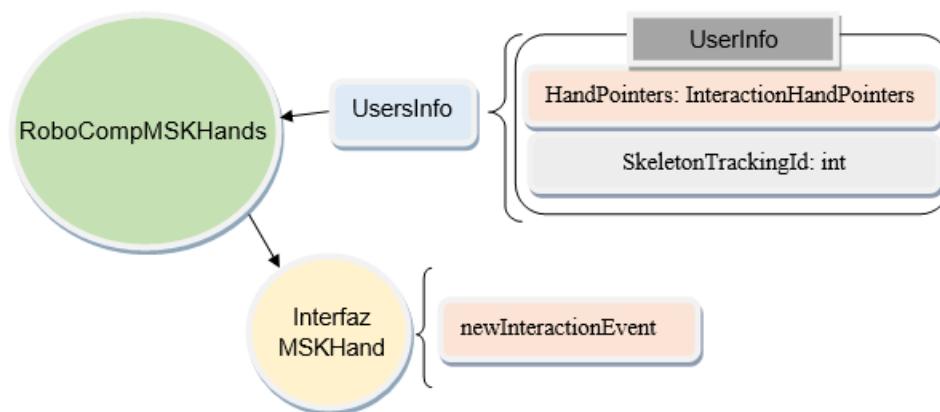


Figura 37. Módulo RoboCompMSKHands del modelo WinKinectComp.

El Módulo RoboCompMSKASR es el encargado de enviar toda la información correspondiente al sonido detectado por el sensor 3D Kinect, el cual tiene un reconocimiento de voz y lanzará este evento cada vez que una palabra sea reconocida (Figura 38). TSentence es una clase que está formada por la frase reconocida (WordRecognizedea), así como la fecha y hora a la que ha sido reconocida.

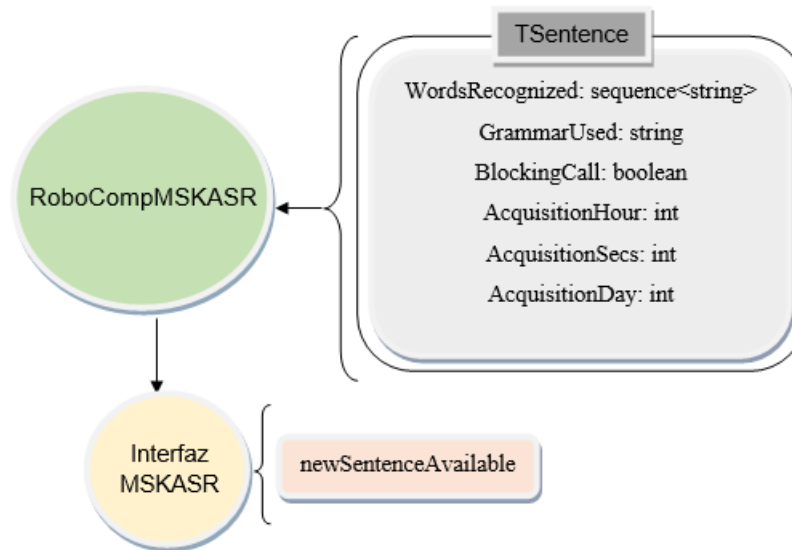


Figura 38. Módulo RoboCompMSKASR del modelo WinKinectComp.



## 3.- *Desarrollo*

Aquí se explicarán detalladamente las tecnologías aplicadas, los casos de uso y las partes que componen el trabajo en cuestión. Se realizará una visión de cada uno de los componentes realizados así como del modelo desarrollado.

### 3.1.- **Análisis**

Este proyecto fin de grado tiene como finalidad la creación de un modelo y dos componentes para que un médico rehabilitador y/o terapeuta pueda añadir nuevas poses y ejercicios al sistema, así como que el sistema pueda reconocer al paciente y poder verificar la correcta realización de los ejercicios propuestos, para ello, tendrá que hacer uso del componente de visión que transformará los datos para que el sistema los reconozca.

Tal y como se vio en el apartado 2.2.- Historia con el proyecto fin de carrera de Santiago Alfaro (Alfaro, 2012), con el componente de visión que estaba implementado en el proyecto NAOTherapist el paciente tenía que colocarse delante del sensor 3D de una forma muy restrictiva para ser reconocido correctamente y que los cálculos que se realizaban para la comparación de poses y el posterior retargeting del robot fuesen correctos. Con el componente de visión desarrollado en este trabajo y explicado más adelante, se pretende dar al sistema más robustez y precisión a la hora de capturar los datos del paciente. Para ello, se ha desarrollado un modelo nuevo que mejora el explicado en el apartado 2.4.2.- WinKinectComp, ya que elimina los datos innecesarios, aumenta los datos proporcionados por el modelo, a la vez que simplifica el sistema, sin perder en ningún momento la retrocompatibilidad con los componentes diseñados hasta la fecha. Con este nuevo componente de visión el paciente tiene más flexibilidad a la hora de colocarse delante del sensor 3D, ya que, como se explicará más adelante, el nuevo componente hace uso de los planos del individuo para el cálculo de los ángulos, lo que permite al modelo y al componente de visión una mayor flexibilidad a la hora de reconocer al paciente. Este nuevo modelo junto con el nuevo componente de visión permite que el componente de aprendizaje de nuevas poses y ejercicios sea más preciso y pueda realizar una mejor comparación de poses debido a que, al calcular los ángulos de las articulaciones del cuerpo del paciente a través de planos, se obtiene una mayor robustez del sistema, ya que permite una mala colocación del paciente y el sistema seguiría funcionando correctamente.

### 3.1.1.- Escenarios

Los escenarios que se exponen a continuación, son aquellos que tienen relación directa con las funcionalidades que se han desarrollado a lo largo de este trabajo fin de grado.

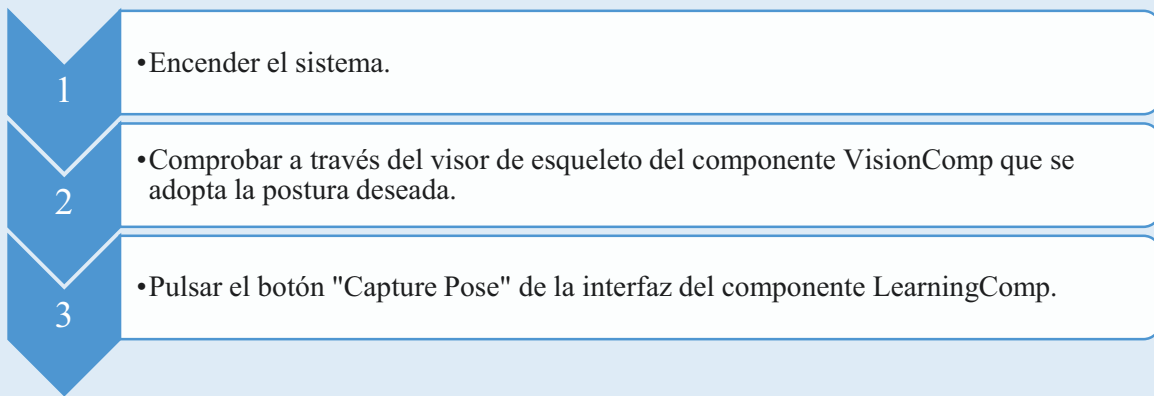
<b>ID:</b> E1	
<b>TÍTULO:</b> Interacción del paciente con el componente VisionComp.	
<b>SITUACIÓN:</b> El paciente al realizar la sesión propuesta por el sistema será reconocido por el componente VisionComp que enviará los datos del paciente a los componentes necesarios para el correcto funcionamiento del sistema.	
<b>METODOLOGÍA:</b>	
1	•Encender el sistema.
2	•Inicia una sesión.
3	•Realizar los ejercicios de la sesión
4	•El sistema corrige las posiciones que están mal, teniendo en cuenta los ángulos formados por las articulaciones del cuerpo del paciente.
5	•Finaliza la sesión.

**ID:** E2

**TÍTULO:** Añadir nuevas poses.

**SITUACIÓN:** El médico rehabilitador y/o el terapeuta necesitan introducir en la base de conocimiento del sistema una nueva pose.

**METODOLOGÍA:**

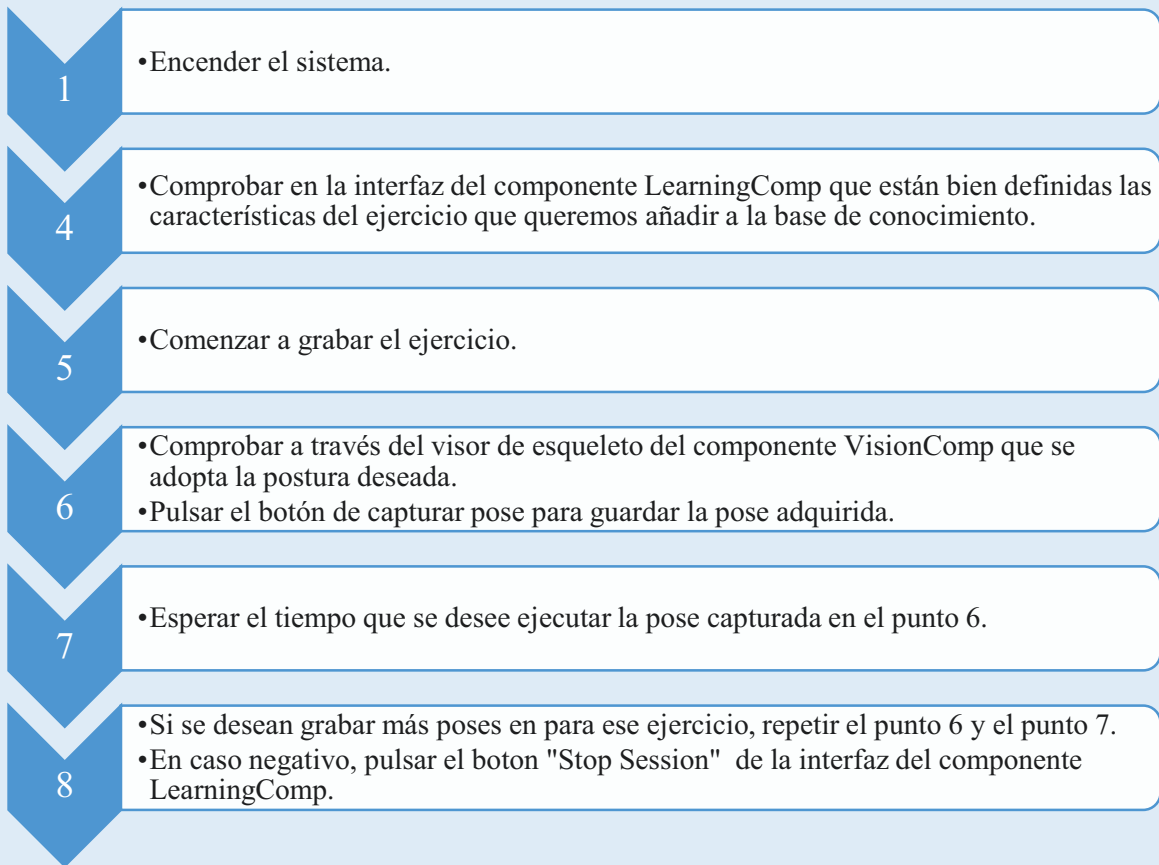


**ID:** E3

**TÍTULO:** Añadir nuevo ejercicio.

**SITUACIÓN:** El médico rehabilitador y/o el terapeuta necesitan introducir en la base de conocimiento del sistema un nuevo ejercicio.

**METODOLOGÍA:**



### 3.1.2.- Casos de Uso

Los casos de uso serán los correspondientes a los escenarios explicados en el apartado anterior.

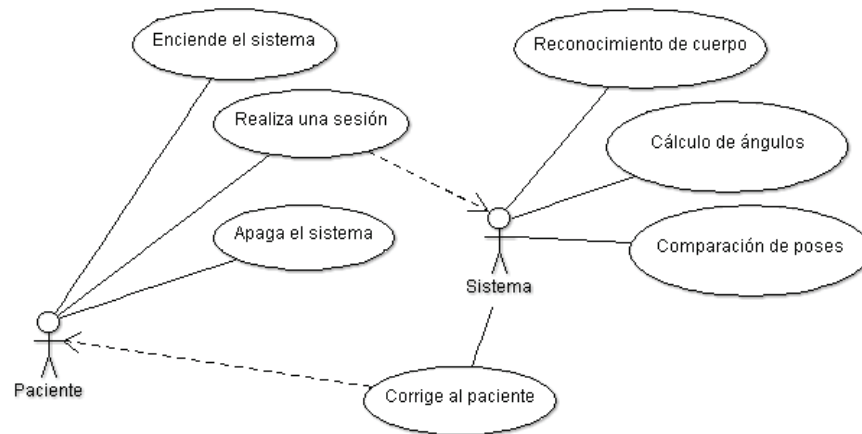


Figura 39. Caso de uso del escenario E1.

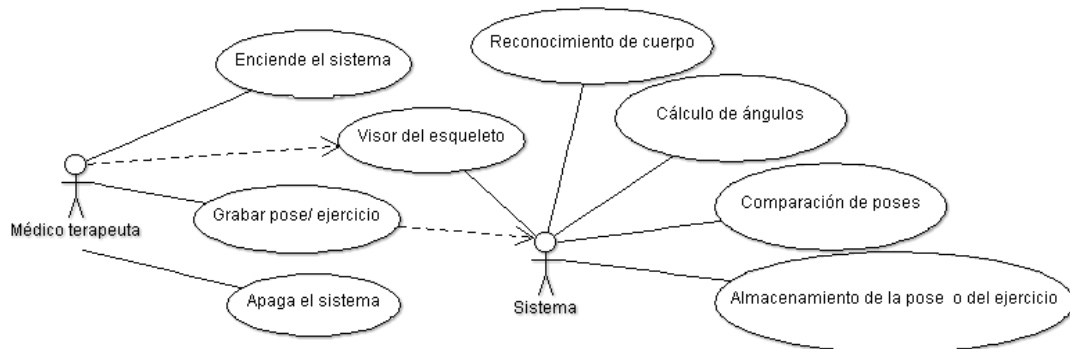


Figura 40. Caso de uso del escenario E2 y E3.

### 3.1.3.- Requisitos funcionales

En esta sección se mostraran una serie de requisitos funcionales que serán necesarios para el correcto funcionamiento del sistema, su puesta en marcha y su correcta utilización.

ID: RF1	Tipo: Funcional	Necesidad: Alta
<b>TITULO:</b> Añadir nuevas poses.		
<b>DESCRIPCION:</b> El terapeuta podrá añadir nuevas poses a la base de conocimiento cuando él lo desee.		

ID: RF2	Tipo: Funcional	Necesidad: Alta
<b>TITULO:</b> Añadir nuevas poses.		
<b>DESCRIPCION:</b> El terapeuta podrá añadir nuevas poses a la base de conocimiento cuando el componente de planificación se lo pida.		

ID: RF3	Tipo: Funcional	Necesidad: Alta
<b>TITULO:</b> Objetivos terapéuticos.		
<b>DESCRIPCION:</b> El paciente puede tener asignada una terapia que sea útil para uno o más de estos objetivos terapéuticos: actividades bimanuales, actividades unimanuales finas, actividades unimanuales gruesas, posicionamiento de mano o posicionamiento de brazo.		

ID: RF4	Tipo: Funcional	Necesidad: Alta
<b>TITULO:</b> Atributos característicos de un ejercicio.		
<b>DESCRIPCION:</b> Los ejercicios disponen de los siguientes atributos: la fase a la que pertenece, estado de si está de pie o sentado, las poses del brazo izquierdo y las del brazo derecho, así como el tiempo que mantiene dichas poses, la intensidad del ejercicio, el grupo de ejercicios al que pertenece, los grados de adecuación del ejercicio y la dificultad del mismo.		

ID: RF5	Tipo: Funcional	Necesidad: Alta
<b>TITULO:</b> Comparación de poses.		
<b>DESCRIPCION:</b> Las poses guardadas serán comparadas con la nueva para determinar si es una pose nueva o ya está almacenada en la base de conocimiento.		

<b>ID:</b> RF6	<b>Tipo:</b> Funcional	<b>Necesidad:</b> Alta
<b>TITULO:</b> Sesiones de rehabilitación variadas.		
<b>DESCRIPCION:</b> El sistema generará sesiones de rehabilitación sin que se repitan ninguno de los ejercicios realizando una sesión que se adecue al paciente.		

<b>ID:</b> RF7	<b>Tipo:</b> Funcional	<b>Necesidad:</b> Alta
<b>TITULO:</b> Adecuación de los ejercicios.		
<b>DESCRIPCION:</b> El médico rehabilitador y el terapeuta indicarán al sistema las restricciones del paciente para que éste pueda realizar unas sesiones de rehabilitación que sean lo más personalizadas posibles.		

<b>ID:</b> RF8	<b>Tipo:</b> Funcional	<b>Necesidad:</b> Alta
<b>TITULO:</b> Evaluación del paciente.		
<b>DESCRIPCION:</b> El sistema evaluará en tiempo real al paciente para comprobar que realiza correctamente la sesión.		

<b>ID:</b> RF9	<b>Tipo:</b> Funcional	<b>Necesidad:</b> Alta
<b>TITULO:</b> Petición de nuevos ejercicios.		
<b>DESCRIPCION:</b> Si el sistema detecta que no dispone de suficientes ejercicios para realizar la sesión se comunicará con el terapeuta para pedirle ejercicios nuevos.		

<b>ID:</b> RF10	<b>Tipo:</b> Funcional	<b>Necesidad:</b> Alta
<b>TITULO:</b> Función espejo.		
<b>DESCRIPCION:</b> El sistema mostrará una interfaz con el esqueleto del paciente en tiempo real para que éste tenga una visión de cómo está realizando el ejercicio.		

<b>ID:</b> RF11	<b>Tipo:</b> Funcional	<b>Necesidad:</b> Alta
<b>TITULO:</b> Corrección del ejercicio.		
<b>DESCRIPCION:</b> Si el sistema detecta que el paciente no está realizando correctamente el ejercicio, le indicará a éste que lo está realizando incorrectamente y le mostrará cómo se realiza.		

<b>ID:</b> RF12	<b>Tipo:</b> Funcional	<b>Necesidad:</b> Alta
<b>TITULO:</b> Movimiento de los brazos del robot.		
<b>DESCRIPCION:</b> El robot debe hacer todas las poses terapéuticas que estén almacenadas en la base de conocimiento.		

<b>ID:</b> RF13	<b>Tipo:</b> Funcional	<b>Necesidad:</b> Alta
<b>TITULO:</b> Presencia de un paciente.		
<b>DESCRIPCION:</b> El sistema tiene que detectar cuando un paciente está situado correctamente para realizar los ejercicios.		

<b>ID:</b> RF14	<b>Tipo:</b> Funcional	<b>Necesidad:</b> Alta
<b>TITULO:</b> Identificación del paciente.		
<b>DESCRIPCION:</b> El sistema tiene que detectar al paciente en caso de que haya más de una persona en la sala.		

## 3.2.- Diseño e implementación

En esta sección, se detallarán las herramientas y tecnologías utilizadas para la realización de este proyecto fin de grado, así como la estructura y funcionalidades del mismo.

### 3.2.1.- Herramientas y tecnologías aplicadas

Para poder realizar este proyecto se han tenido que utilizar una serie de tecnologías y herramientas. Este proyecto se ha realizado en Python y en Ice, ya que es un proyecto fin de grado que forma parte del proyecto NAOTherapist de la Universidad Carlos III de Madrid en el cual se utilizan dichas tecnologías.

Para el manejo de los datos proporcionados por el componente WinKinectComp se ha tenido que usar varias librerías, tales como:

- Ice: Librería de ZeroC que se encarga de la comunicación entre componentes, como: VisionComp, LearningComp y WinKinectComp.
- Pyglet<sup>16</sup>: Es una librería orientada al desarrollo de aplicaciones en Python visuales, por ello está basada en OpenGL y GLU. No tiene dependencias, por lo que con solo instalarlo en el sistema y

---

<sup>16</sup> [www.pyglet.org/](http://www.pyglet.org/)



teniendo instalado Python bastaría para hacerlo funcionar. Es una librería muy optimizada para Python por lo que produce el máximo rendimiento a la hora de realizar las imágenes renderizadas. Genera las imágenes en tiempo real de los datos proporcionados por el componente WinKinectComp.

- Xml: Librería interna de Python que sirve para el manejo de archivos xml.
- PyQt 4<sup>17</sup>: Es una librería construida sobre el framework de la plataforma QT. Para el uso de esta librería es necesario tener instalado en el ordenador QT v4. Se ha desarrollado sobre esta versión y no sobre la última versión, porque el proyecto tenía que realizarse sobre Python 2.7 y la versión 5 de QT funciona sobre Python 3.4. Se encarga de la interfaz gráfica del componente LearningComp.

### 3.2.2.- Estructura del proyecto

En este apartado se explicará brevemente cuál es la estructura y cómo funciona el proyecto.

El proyecto está formado por dos componentes:

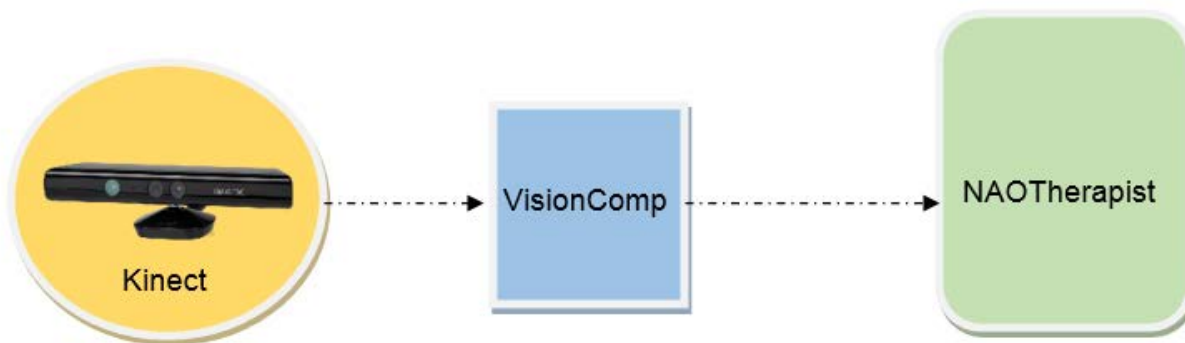
- VisionComp: Obtiene los datos del componente WinKinectComp, los procesa y rellena el modelo RoboCompKinectModel. Se convierte en publicador Ice y envía a los componentes que se lo soliciten el modelo RoboCompKinectModel a través de sus interfaces. Sirve como visor, ya que muestra una reconstrucción del esqueleto de persona que capta la Kinect.
- LearningComp: Interfaz gráfica mediante la cual el médico y/o terapeuta graba las nuevas poses o ejercicios completos. Se conecta al componente VisionComp y se relaciona con las bases de datos.

### 3.2.3.- Funcionamiento

Lo primero que se tuvo que hacer fue estudiar qué datos proporciona el componente WinKinectComp y cómo los envía al resto de componentes. Se descubrió que los datos los envía a través de una interfaz Ice utilizando IceStorm. También está preparado para enviar la imagen RGB y de profundidad, pero estos métodos no están implementados, por lo que este componente solo envía los módulos del cuerpo, la cara, las manos y el reconocimiento de la voz (Figura 41).

---

<sup>17</sup> <http://www.riverbankcomputing.co.uk/software/pyqt/intro>



*Figura 41. Relación Kinect - VisionComp – NAOTherapist.*

Una vez se supo cómo funcionaba el componente WinKinectComp, se investigó cómo implementar la interfaz Ice en Python para poder crear cualquier componente que recibiese los datos que enviaba el componente WinKinectComp. Sabiendo esto, se creó desde cero el componente VisionComp, el cuál recibe los datos antropométricos y los preprocesa para poder crear modelo del cuerpo adaptado a las necesidades del proyecto. Este modelo se sirve a petición de los componentes que se lo soliciten a VisionComp, de modo que todo el procesamiento sobre ángulos se realiza en este módulo. Por lo tanto el componente VisionComp realizaba la función de cliente Ice servidor obteniendo los datos del componente WinKinectComp y de servidor Ice enviando los datos procesados a los componentes que los soliciten a través del módulo Ice desarrollado en este proyecto.

Cuando el componente VisionComp estuvo acabado, se implementó un nuevo componente que fue el LearningComp que es el encargado de añadir nuevas poses y ejercicios a la base de conocimiento y que de esa forma el robot pueda crear mejores sesiones de rehabilitación.

El componente LearningComp se conecta al componente VisionComp a través de Ice para obtener los datos que envía y que permite al usuario poder guardar nuevas poses de forma individual o realizar una grabación de un ejercicio.

### **3.2.4.- Modelo de VisionComp**

Para este proyecto se ha decidido crear un modelo que ampliase los datos que recibía del componente WinKinectComp, para que fuese lo más completo y preciso posible y, a la vez, pudiese ser usado por otras personas más adelante sin tener que modificar el componente, ya que aporta todo lo necesario porque, aunque hay partes que no se utilizan en este proyecto, tales como el reconocimiento de voz, caras y manos, podrían ser usadas en un futuro proyecto. En este apartado se explicará la estructura del modelo así como las decisiones que se han tomado.

### 3.2.4.1.- Estructura del modelo

En esta sección se detallará la estructura general del modelo, que será adjuntado completo en el apartado 7.2.1.- Módulo RoboCompKinectModel.

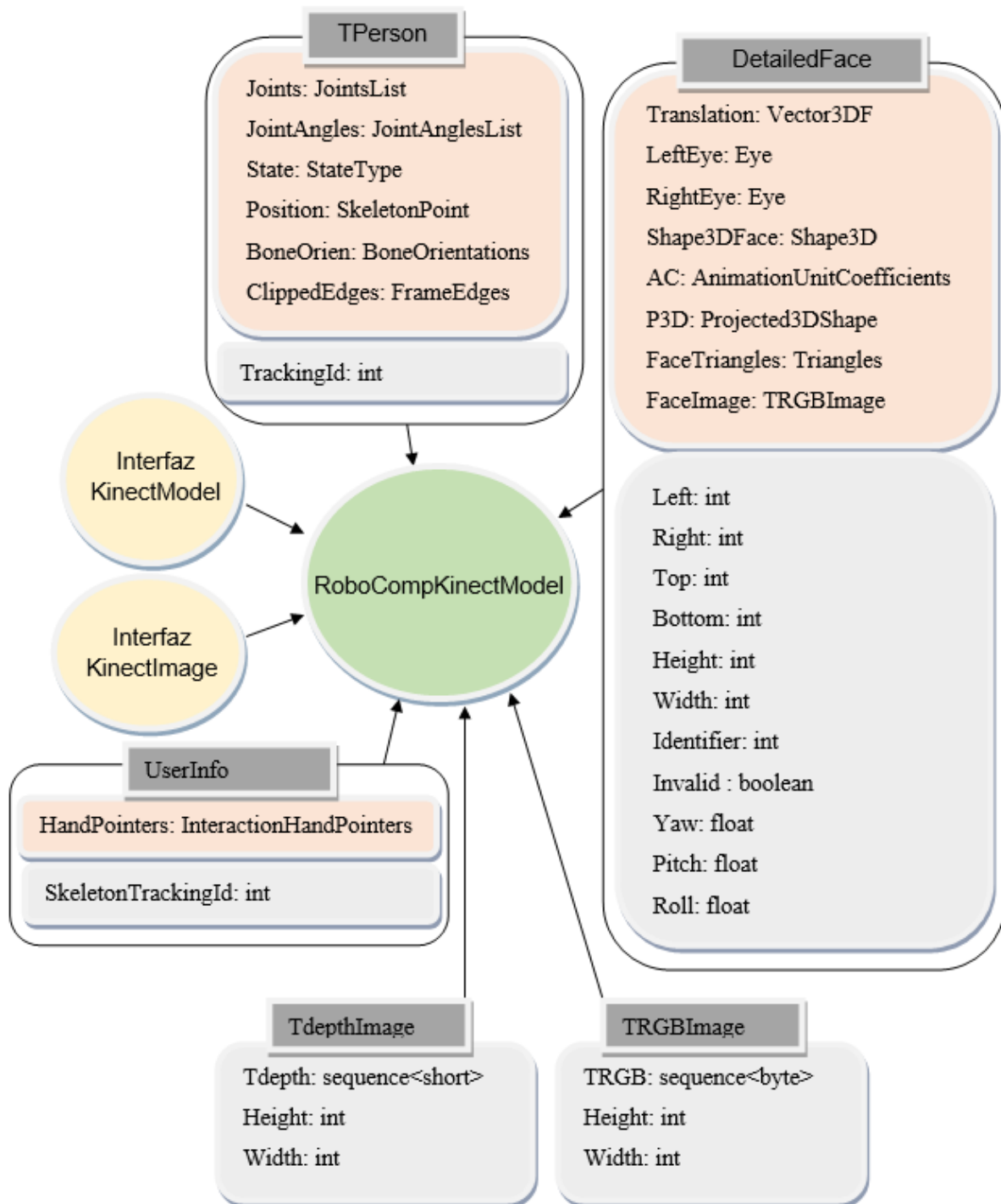


Figura 42. Modelo RoboCompKinectModel.

### 3.2.4.2.- Modificaciones del modelo

Como puede observarse al comparar el modelo de WinKinectComp (Apartado 2.4.2.- WinKinectComp) y el modelo RoboCompKinectModel (Figura 42), la estructura del nuevo modelo no afecta al correcto funcionamiento de los componentes creados hasta la fecha, ya que mantiene la retrocompatibilidad con componentes antiguos y aumenta la información enviada para los componentes nuevos que quieran utilizarla. Como ya se comentó en otra parte de esta memoria, parte del componente WinKinectComp que no se ha utilizado, ha sido introducido exactamente como estaba para que futuros proyectos tuviesen la información, como puede verse en la Figura 43.

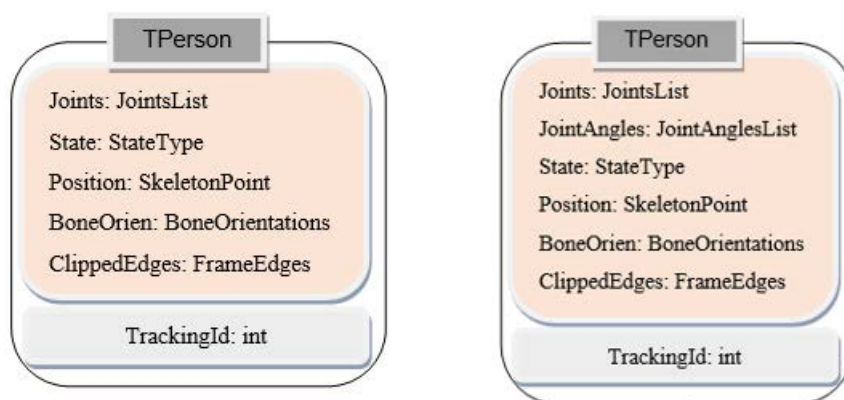


Figura 43. Clase TPerson: antigua (izquierda) y nueva (derecha).

Con el objetivo de facilitar la utilización por otros componentes, se decidió unificar toda la información referente a las personas que capturaba la Kinect en un solo modelo de tal forma que con realizar una sola conexión al módulo Ice, se pudiese recoger todo lo necesario (Figura 44), y de esta forma, separar la parte de envío de video que no se consiguió que funcionase correctamente, como ya se explicó anteriormente, del resto de información, mediante dos interfaces diferentes, una para los datos y otra para las imágenes.

El principal dato que se utiliza en este proyecto es el cuerpo que se recoge de la Kinect, con una serie de modificaciones necesarias para poder tratar esos datos en el componente LearningComp, por lo que el componente VisionComp procesa los datos recibidos del componente WinKinectComp y los transforma rellenando el componente RoboCompKinectModel enviándolos a través de Ice. Por lo tanto, los datos recibidos del componente WinKinectComp con la información de la cara y las manos, son enviados sin ningún tipo de modificación, y el módulo de reconocimiento de voz ha sido eliminado completamente (Figura 42).

La interfaz KinectImage es la encargada de enviar los datos de las imágenes recibidas a través del componente WinKinectComp sin tratar. Esta interfaz dispone de dos métodos (Figura 44):

- getDepthImage: Devuelve la imagen de profundidad.

- `getRGBImage`: Devuelve la imagen RGB.



*Figura 44. Interfaces del Modelo RoboCompKinectModel.*

A pesar de que estas interfaces no funcionan, debido a que el componente `WinKinectComp` no rellena estos datos de la Kinect, se han introducido por si se arreglase ese problema en el componente `WinKinectComp`.

La interfaz `KinectModel` es la encargada de enviar los datos de las personas recibidas a través del componente `WinKinectComp`, tratando los datos para que sean como se ha decidido en el modelo utilizado en este proyecto. Esta interfaz dispone de tres métodos (Figura 44):

- `getFace`: Devuelve los datos de la cara.
- `getHands`: Devuelve los datos de las manos.
- `getBody`: Devuelve los datos del cuerpo.

El método `getBody` elimina los datos que se han considerado innecesarios del módulo `RoboCompMSKBody` del componente `WinKinectComp`, como son los datos sobre los cuaterniones, y se han completado los datos recibidos con los datos correspondientes a los “`JointAngles`”, que son los ángulos formados por los joints que son necesarios para la comparación de poses por el componente `LearningComp` (Figura 45).



Figura 45. Código añadido por `VisionComp` al modelo de `WinKinectComp`.

`JointAngles` es una lista formada por un “`AngleType`” y por un “`jointAngle`”. El `AngleType` es el tipo de ángulo formado, en el cual hay una lista definida en el modelo de los posibles valores que puede tener este campo, y el `jointAngle` es una estructura de datos que está formada por: `startSegment` (segmento inicial), `endSegment` (segmento final), y el cálculo del seno y coseno formado por dichos segmentos.

Los segmentos corresponden con los huesos y los planos: coronal, transversal y sagital, que son los que han sido definidos en el modelo usado (Figura 46).

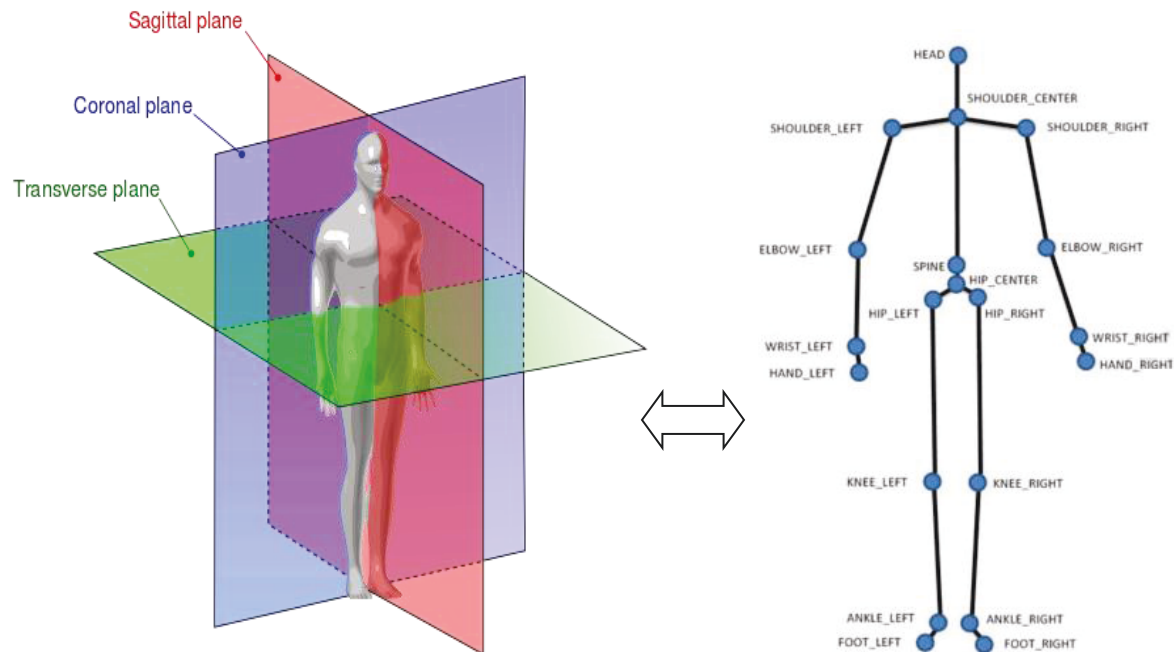


Figura 46. Planos del individuo y sus articulaciones y huesos.

De esta forma, se mantiene la estructura original añadiendo datos que serán necesarios para desarrollar los componentes que forman este proyecto.

### 3.2.5.- VisionComp

VisionComp recibe del componente WinKinectComp un modelo de datos separados por cuerpo, cara, manos y reconocimiento de voz, los cuales se envían de forma asíncrona y mediante eventos, es decir, cada uno se envía sólo cuando es detectado por el componente WinKinectComp, con una tasa máxima de 30fps, y no de forma continua, éste envía los frames con la información lo más rápido posible y cada vez que procesa un dato, envía un evento que tiene que ser captado por el componente que quiera recibir dicha información.

### 3.2.5.1.- Partes

El componente WinKinectComp envía la información del cuerpo, manos, cara y voz por separado y de forma asíncrona, según obtiene la información, crea el evento. Por lo que se tienen que capturar y analizar cada parte por separado.

#### 3.2.5.1.1.- Body

VisionComp analiza la lista de cuerpos que le ha enviado WinKinectComp y realiza una comparación entre ellos para determinar qué cuerpo es al que va a prestar atención. Esta comparación la realiza obteniendo la distancia a la que están los joints ShoulderCenter de las personas detectadas, y el que esté más cerca es la persona en la que se centrará VisionComp realizar el resto de operaciones.

Este paso previo se realiza como medida de seguridad en el supuesto en el que haya más de una persona en la sala. Una vez se ha localizado el sujeto del cual se van a realizar las operaciones, se dispone a obtener las posiciones de sus joints (x,y,z) y su estado (Tracked, Inferred, notTracked).

Con las posiciones de los joints, se calculan los vectores que forman los joints que están unidos entre sí, conforme la anatomía del cuerpo humano.

Se calculan los planos del cuerpo humano de la siguiente forma para que puedan ser reconstruidos en otros componentes y así poder reconstruir el esqueleto de la misma forma que lo hace el componente VisionComp y obtener los mismos ángulos.

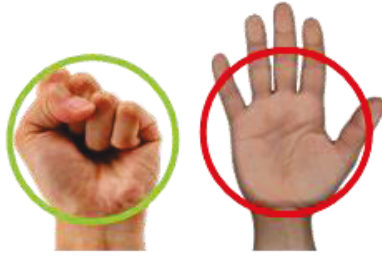
- Plano coronal: Está formado por la unión de las articulaciones: ShoulderRight, ShoulderLeft y HipCenter.
- Plano transversal: Está formado por el plano coronal y las articulaciones ShoulderRight y ShoulderLeft.
- Plano sagital: Está formado por la intersección de los planos transversal y coronal.

A continuación, se obtienen los ángulos que forman cada uno de los vectores con los planos y/o con los vectores adyacentes, en caso de que sean necesarios. Para cada uno de los ángulos se calcula el coseno y el seno formado por los vectores, tal y como se desarrolla en la sección 3.2.4.2.- Modificaciones del modelo.

#### 3.2.5.1.2.- Hands

Cuando se recibe el evento de manos desde el WinKinectComp se comprueba que sea de la misma persona de la que se están obteniendo los datos del cuerpo y se verifica el estado de cada una de las manos (GripRelease, Grip, NoneEvent). Si el estado es GripRelease o NoneEvent, se rellena el modelo como seno igual a 0 y coseno igual a 1, en caso contrario seno es 1 y coseno es 0 ( Figura 47).

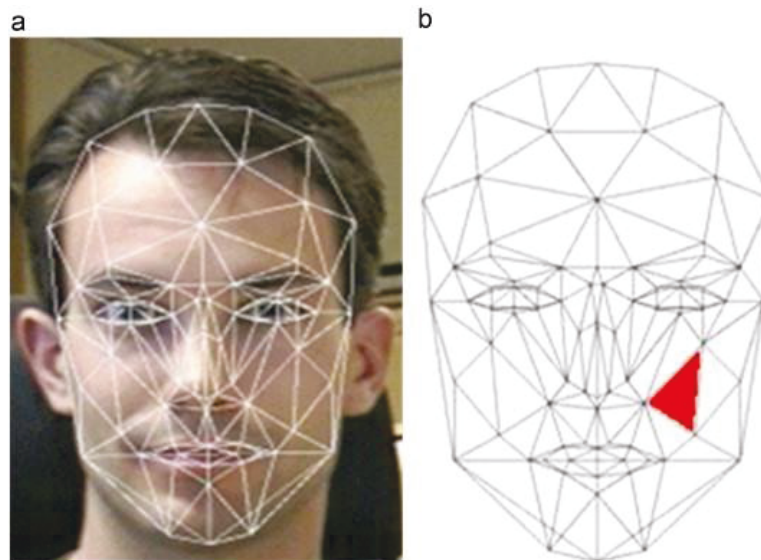




*Figura 47. Cálculo de la apertura de la mano.*

### 3.2.5.1.3.- Face

Este proyecto no utiliza esta función pero como se quería hacer un modelo completo, se incorporó en el modelo directamente de la forma en la que se recibe del componente WinKinectComp, por si se quisiera continuar con esta parte del modelo en un futuro (Figura 48).



*Figura 48. Proyección de la malla de puntos de la cara.*

### 3.2.5.1.4.- Voice

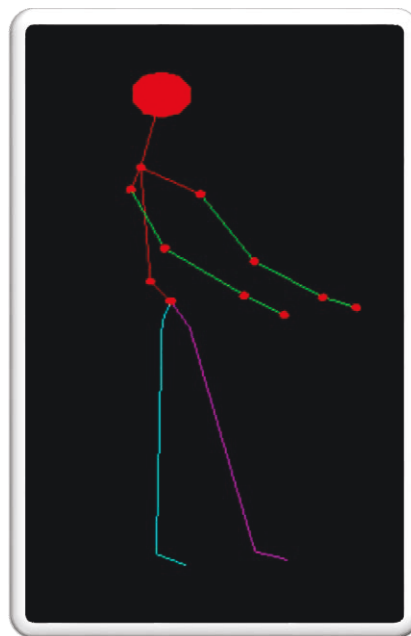
En un principio se intentó el uso de esta función dentro del proyecto, pero el componente WinKinectComp es muy lento a la hora de reconocer palabras por lo que se hacía inviable el uso en tiempo real de la aplicación y no se añadió la funcionalidad al modelo.

### 3.2.5.2.- Imagen en tiempo real

Tal y como se ha comentado anteriormente, el componente WinKinectComp no tiene implementado el envío de la imagen RGB y de profundidad por lo que no se podían utilizar directamente los datos para mostrarlos en VisionComp sin tener que procesar la información.

Como la principal función del video en el componente VisionComp es tener una idea de cómo la Kinect estaba reconociendo a la persona que interactuaba con ella, se decidió que haciendo una reconstrucción del esqueleto mediante la unión de los joints, anatómicamente, identificando la posición de los joints y la cabeza, el usuario tendría una visión en tiempo real de la imagen que estaba obteniendo la Kinect y de esa forma podría comprobar si la postura que está realizando es reconocida correctamente por el sistema.

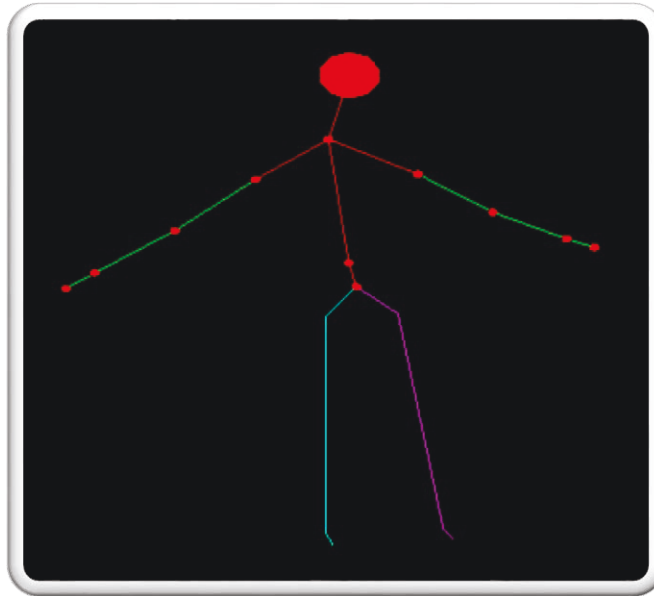
Para la reconstrucción del esqueleto se hizo uso de la librería Pyglet que se basa en OpenGL. Una vez normalizados los datos, el componente VisionComp realiza como máximo 30 imágenes por segundo. En cada imagen, une los joints de forma anatómica, como huesos, señala los joints con esferas y crea una esfera más grande para indicar la cabeza (Figura 49).



*Figura 49. Reconstrucción del cuerpo con Pyglet.*

#### **3.2.5.2.1.- Normalización**

Las posiciones de las articulaciones obtenidas desde WinKinectComp son las distancias que existen entre la articulación y la Kinect, por lo que para poder obtener un esqueleto correctamente formado se tuvieron que normalizar dichos valores. Se tomó la decisión de colocar la articulación del tronco (Spine) como punto origen (0,0,0) y calcular las posiciones del resto de las articulaciones respecto a este. De esta forma, como puede verse en la Figura 50, se realiza una reconstrucción correcta.



*Figura 50. Reconstrucción del cuerpo normalizado.*

#### **3.2.5.2.2.- Pruebas previas del tratamiento de la imagen**

En un principio, se decidió modificar el componente WinKinectComp para que enviase los datos correspondientes a la imagen obtenida por el sensor 3D, ya que tenía implementadas algunas partes y se pensó que completando dicha implementación se podría mandar las imágenes a través de Ice y su posterior utilización en los componentes que lo solicitaran.

Una vez corregido el código de WinKinectComp para que enviase los datos, se implementó en el componente VisionComp la reconstrucción de la imagen mediante la librería OpenCV y NumPy, las cuales son unas librerías muy eficientes y que obtienen el máximo rendimiento. La imagen obtenida tenía un retraso de más de 3 segundos, lo que hacía imposible su uso en tiempo real.

#### **3.2.6.- LearningComp**

Es la interfaz gráfica que se encarga de la grabación de nuevas poses y ejercicios dentro de la base de conocimiento para que el componente Decision Support pueda acceder a esos nuevos datos.

La interfaz gráfica de LearningComp está desarrollada con la librería PyQt4, ya que es una de las librerías más extendidas dentro de Python y al desarrollarse en QT, el código puede ser portado a diferentes lenguajes de programación, porque QT es multiplataforma.

##### *3.2.6.1.- Interfaz gráfica*

La interfaz proporciona todas las funciones y campos necesarios para que el usuario pueda grabar poses y ejercicios completos (Figura 51).

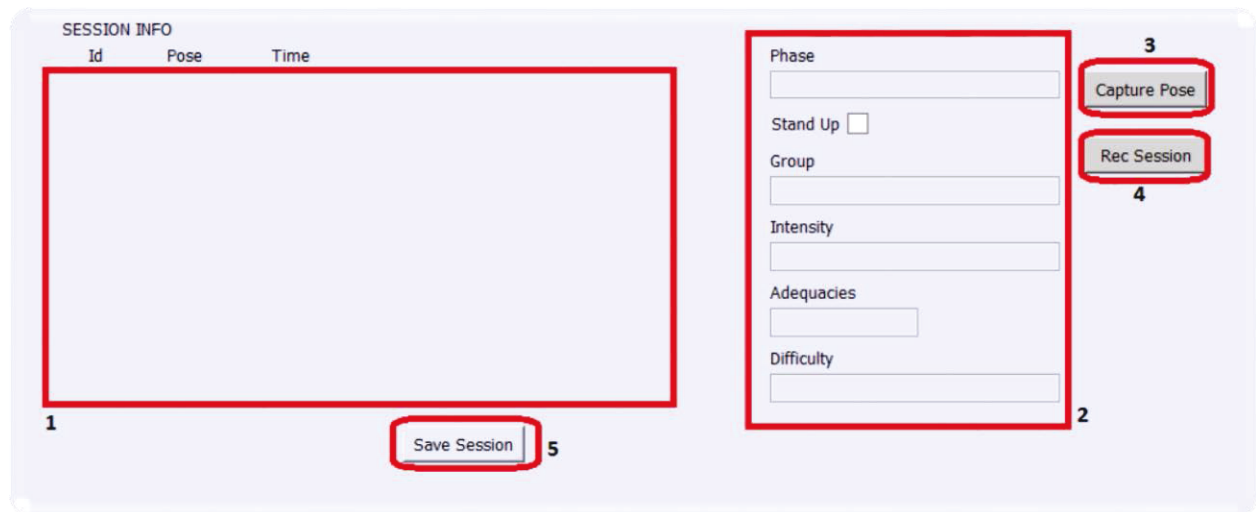


Figura 51. Interfaz gráfica del LearningComp.

El recuadro 1, es un área en el cual se muestran los ejercicios que se han grabado durante la sesión. Los campos del recuadro 2, son las características que tiene el ejercicio que se va a grabar. Estos campos pueden ser completados por el usuario o pueden venir predefinidos por el planificador, dependiendo del tipo de ejercicio que necesita en su base de conocimiento, en base a estas características, el planificador de alto nivel determinará los ejercicios que compondrán las sesiones (González, 2014).

Características del ejercicio:

- Phase: calentamiento, entrenamiento y enfriamiento.
- Stand Up: Indicar si el paciente debe estar levantado para realizar este ejercicio.
- Group: El grupo está relacionado con las capacidades motoras que el paciente necesita para poder realizar el ejercicio.
- Intensity: Está asociado al ritmo cardíaco medio cuando se realiza el ejercicio.
- Adequacies: El nivel de adecuación para cada objetivo terapéutico.
- Difficulty: Nivel de dificultad de un cierto paciente para realizar el ejercicio.

Botones:

- Capture Pose: Con este botón se captura la pose que se quiera capturar y guarda una imagen del esqueleto de la pose. Cuando se presiona el botón, se abre una ventana con una cuenta atrás de 3 segundos y captura la pose.
- Rec Session: Este botón se encarga de grabar los ejercicios completos. Cuando se presiona el botón, éste comienza a parpadear en rojo y cada vez que se presiona el botón "Capture Pose", se realizará una instantánea de la pose que se esté tomando e iniciará un temporizador para saber

cuánto tiempo se mantiene dicha pose. Una vez finalizada la sesión, se vuelve a pulsar el botón “Rec Session”, el cual, parará la grabación y mostrará la sesión grabada en el recuadro (1).

- Save Session: Este botón guardará la sesión grabada en la base de conocimiento para que el planificador disponga de este ejercicio a la hora de gestionar las sesiones.

### 3.2.6.2.- Captura de poses

A continuación, se explicará de una forma más detallada cual es el funcionamiento del botón “Capture Pose” de la interfaz gráfica del componente LearningComp.

Lo primero que sucede al presionar el botón es que el programa comprueba si la función grabar ejercicio ha sido presionada con anterioridad o si por el contrario solamente se quiere grabar una pose. En caso de que la función de grabado haya sido presionada se explicará más adelante los procesos. En el supuesto en el que solo se quiera capturar una pose, se abrirá una ventana con un contador con una cuenta atrás de 3 segundos (Figura 52).

Una vez pasado ese tiempo, el componente solicitará a través de la interfaz KinectModel del módulo RoboCompKinectModel al componente VisionComp que le envíe los datos de la persona que tiene en ese momento delante, llamando al método `getBody`. Con el cuerpo devuelto del método `getBody`, se separan los ángulos de los joints del brazo derecho y los del izquierdo, y la posición de los joints y se guardan en variables para que sean usados más adelante.

A continuación, se realiza la comparación entre la pose que se recibe del componente VisionComp y las que se tienen almacenadas en la base de datos. Se compara cada brazo por separado, ya que en la base de datos están separados. En caso de que cualquiera de los brazos, también llamado chains, no estén en la base de conocimiento, la pose de ese brazo es guardada y se mostrará una captura del esqueleto, reconstruido de la misma forma que en el componente VisionComp, también se guarda en la base de datos la posición de los joints.

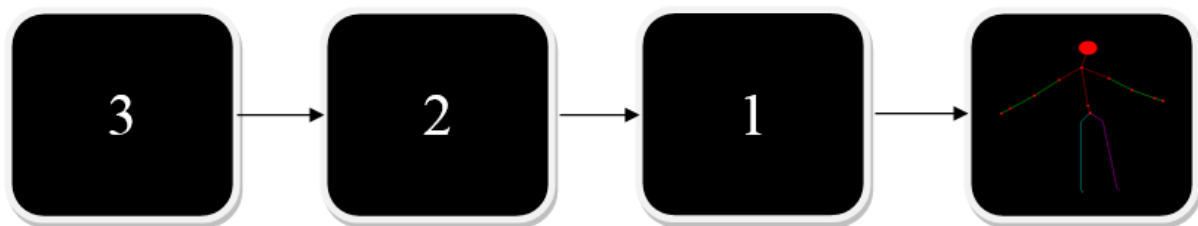
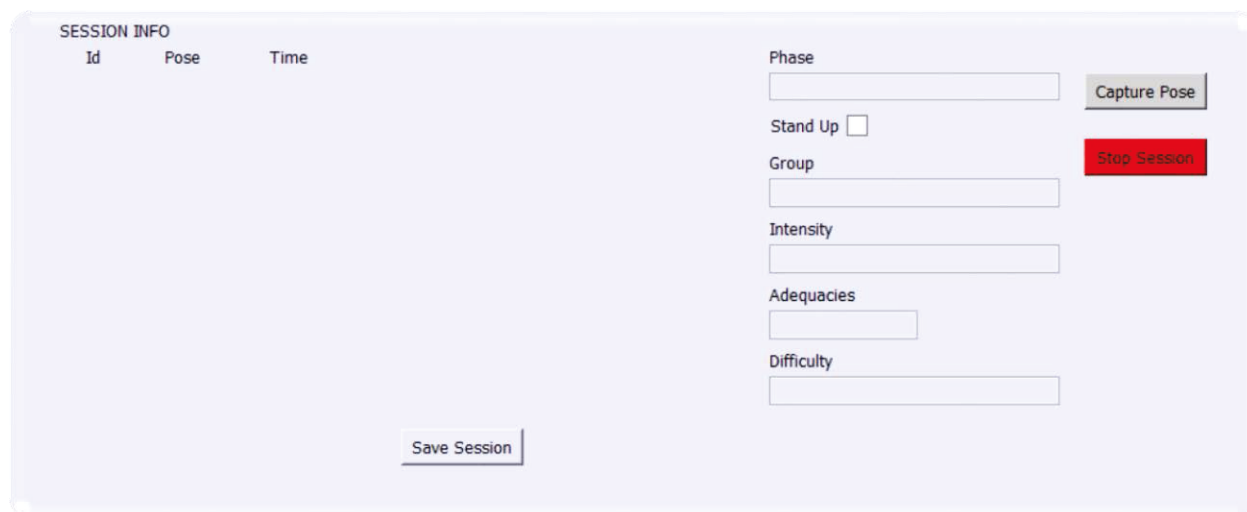


Figura 52. Captura de nueva pose.

### 3.2.6.3.- Captura de ejercicios

En este apartado se explicará de forma detallada el funcionamiento de los procesos ejecutados para la correcta grabación de ejercicios para que el planificador los use.

Cuando se pulse el botón “Rec Session”, éste se pondrá a parpadear en rojo, indicando que la sesión ha comenzado y cambiando el texto del botón a “Stop Session” (Figura 53).



The screenshot shows a web interface for capturing a session. On the left, under the heading "SESSION INFO", there is a table with three columns: "Id", "Pose", and "Time". The table is currently empty. On the right side, there are several input fields and buttons. At the top right is a "Capture Pose" button. Below it is a "Stand Up" checkbox. Further down are input fields for "Group", "Intensity", "Adequacies", and "Difficulty". At the bottom center is a "Save Session" button. On the right side, below the "Capture Pose" button, there is a red button labeled "Stop Session".

Figura 53. LearningComp grabando una sesión.

Para grabar una pose en fase de grabación se tendrá que pulsar el botón “Capture Pose”. El programa captura la pose de idéntica forma que al pulsar el botón sin estar grabando, con la diferencia de que mide el tiempo que pasa entre que se graba una pose y la siguiente, para conocer el tiempo que se mantiene una pose. Al presionar el botón de “Stop Rec”, el recuadro 1 de la interfaz gráfica se rellena con las poses que están listas para ser grabadas y los tiempos que se ha mantenido cada una de ellas. Se tiene que presionar el botón “Save Session” para que el ejercicio se guarde en la base de datos.

Cuando se termina de grabar el ejercicio se rellena el recuadro 1 (Figura 51) con los datos del ejercicio. En el recuadro se informa del identificador de la línea, el ejercicio que ha realizado (unión de la pose izquierda y derecha) y el tiempo que ha mantenido dicho ejercicio. Se dispondrá de dos botones con los que poder ver la pose guardada y para borrar dicha pose si no es la deseada (Figura 54).

SESSION INFO

Id	Pose	Time		
0	r2-l10	7	Show Pose	Delete Pose
1	r3-l11	7	Show Pose	Delete Pose
2	r10-l5	5	Show Pose	Delete Pose

Save Session

Phase

Capture Pose

Stand Up ☐

Group

Rec Session

Intensity

Adequacies

Difficulty

Figura 54. LearningComp con una sesión generada.

Presionando el botón “Show Pose” que se encuentra al lado de cada una de las poses guardadas nos abrirá una ventana con la reconstrucción del esqueleto realizando la pose y de esta manera poder tener una idea de la sesión que se ha grabado (Figura 55). Otro botón que se dispone es el botón “Delete Pose” el cual al ser presionado, eliminará esa pose de la sesión.

Todos los botones deben ser presionados por un ayudante o con un ratón inalámbrico o algún periférico similar para poder realizar la acción y presionar los botones sin interferir en la grabación.

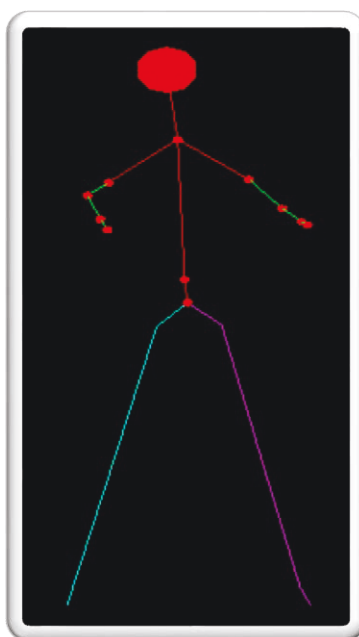


Figura 55. Botón Show Pose del LeaningComp.

#### 3.2.6.4.- Comparación de poses

Una vez obtenido el cuerpo del componente VisionComp, se recoge la información de los ángulos almacenados en la variable `jointAngles` y se realiza una comparación directa con cada una de las poses guardadas en la base de conocimiento. Si la distancia entre alguno de los ángulos del cuerpo y los ángulos de la base de datos es mayor de  $10^\circ$ , se considera que es una nueva postura (Figura 56).

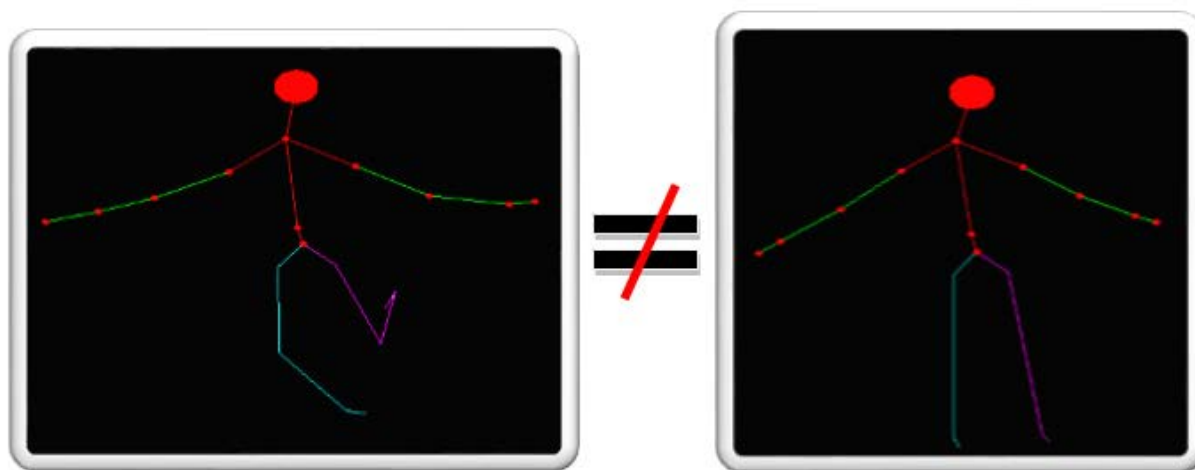


Figura 56. Comparación de poses.

#### 3.2.7.- Estructura de la base de conocimiento

La base de datos utilizada para el proyecto son una serie de archivos en XML, ya que es lo que se ha utilizado en el proyecto NAOTherapist del cual forma parte este proyecto. Para esta parte del proyecto se necesitan cuatro archivos XML. Dos de ellos serán los encargados de almacenar los datos correspondientes a los miembros superiores de forma separada (Figura 57), mientras que otro de los archivos almacenará los datos de las diferentes posiciones tridimensionales de las articulaciones del cuerpo (Figura 58), y el último fichero almacenará los ejercicios compuestos por poses que se vayan almacenando para poder planificar las sesiones de rehabilitación (Figura 59).



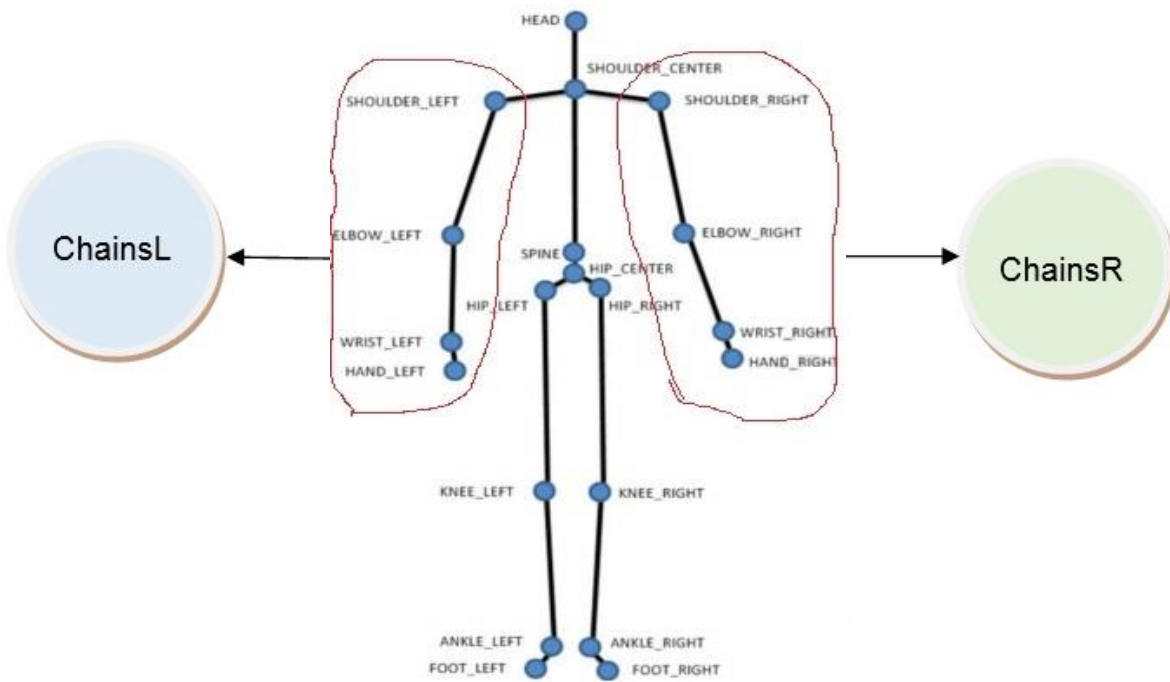


Figura 57. Almacenamiento de los ángulos de cada brazo en la base de conocimiento.

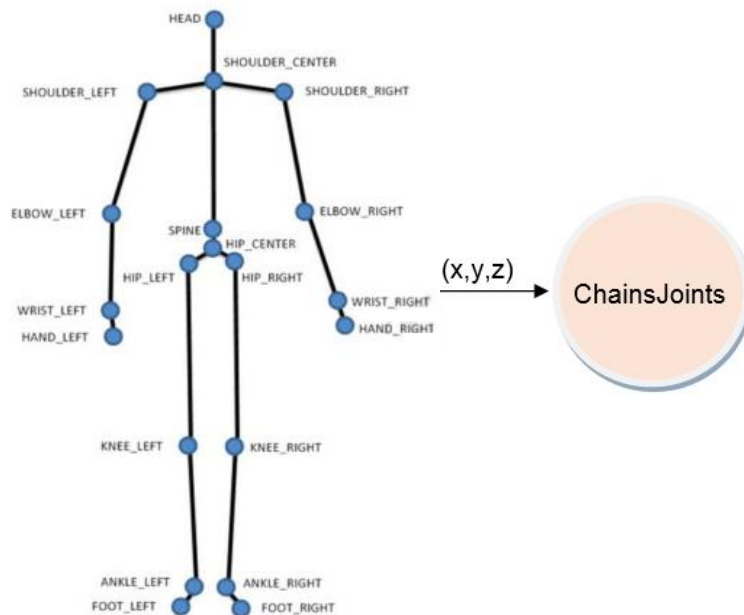
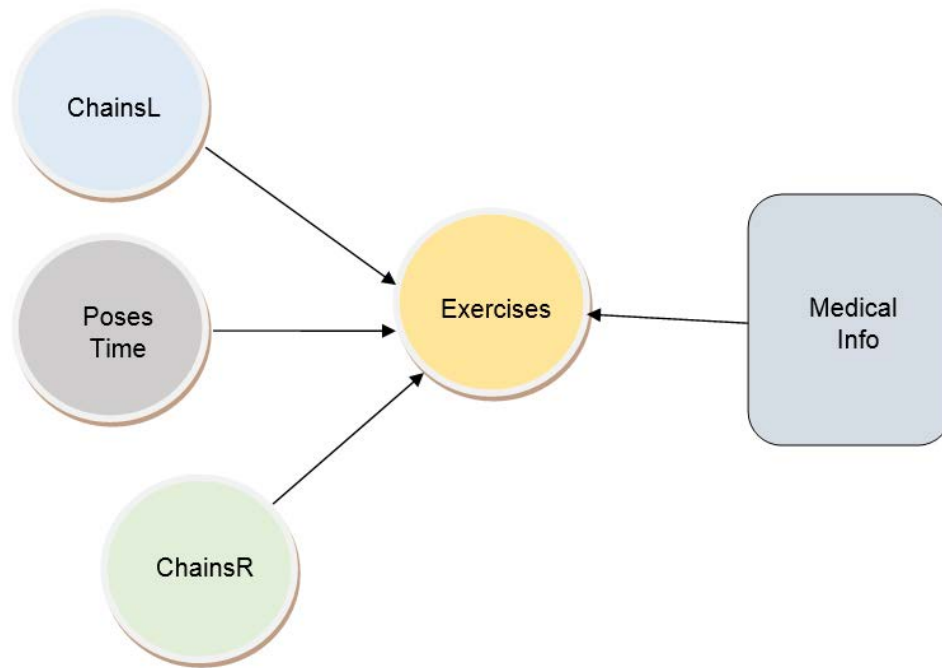


Figura 58. Almacenar posición de cada articulación del cuerpo en la base de conocimiento.



*Figura 59. Almacenamiento de los ejercicios en la base de conocimiento.*

#### *3.2.7.1.- ChainsL y ChainsR*

Son los ficheros encargados de almacenar los ángulos formados por los brazos. Cada uno está compuesto por las poses correspondientes a su brazo y cada pose se compone de los senos y cosenos de los ángulos que forman cada una de las articulaciones con los planos sagital, coronal y transversal o entre las articulaciones, dependiendo de cuáles sean útiles para la comparación de poses. Por lo que el ChainsL se formaría por los ángulos formados por las articulaciones del brazo izquierdo (Figura 60) y el ChainsR por los del brazo derecho (Figura 61).

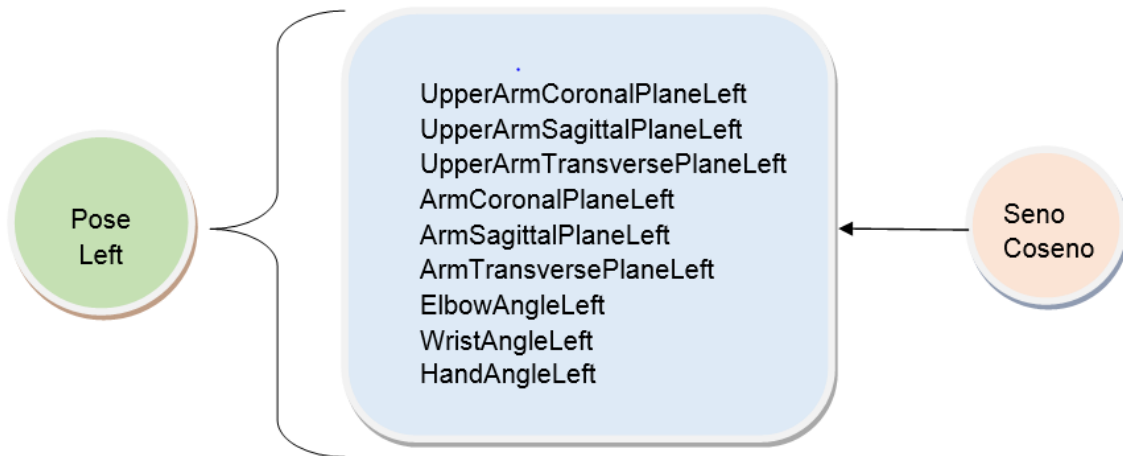


Figura 60. Estructura ChainsL.

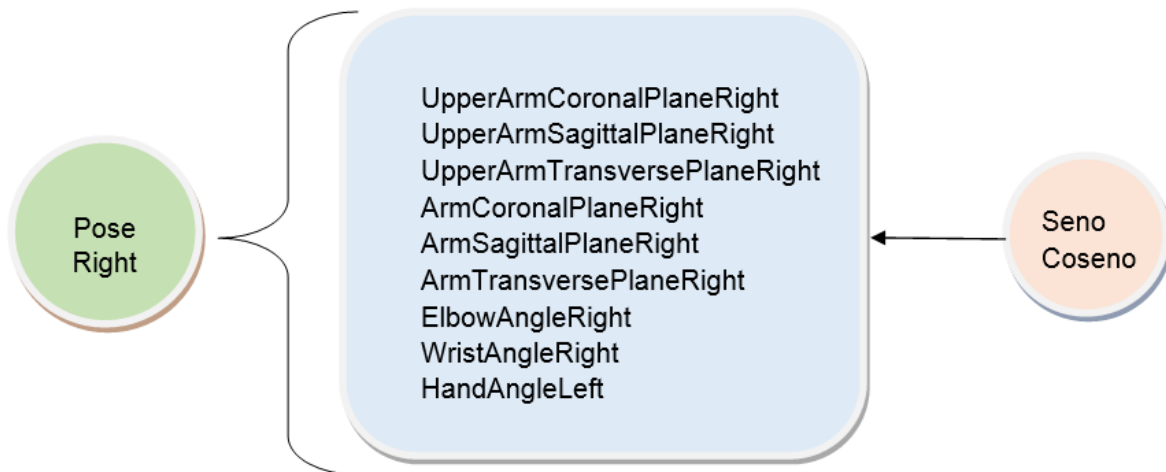


Figura 61. Estructura ChainsR.

### 3.2.7.3.- ChainsJoint

Es el encargado de almacenar las posiciones de las articulaciones del cuerpo entero para una posterior reconstrucción del esqueleto del paciente y/o para un posterior procesamiento de los datos. Cada postura recibe el nombre de la unión de las posturas del brazo izquierdo y el brazo derecho que tengan en el momento de capturar los datos y se compone de las posiciones (x,y,z) de cada articulación tal y como se reciben del componente VisionComp (Figura 62).

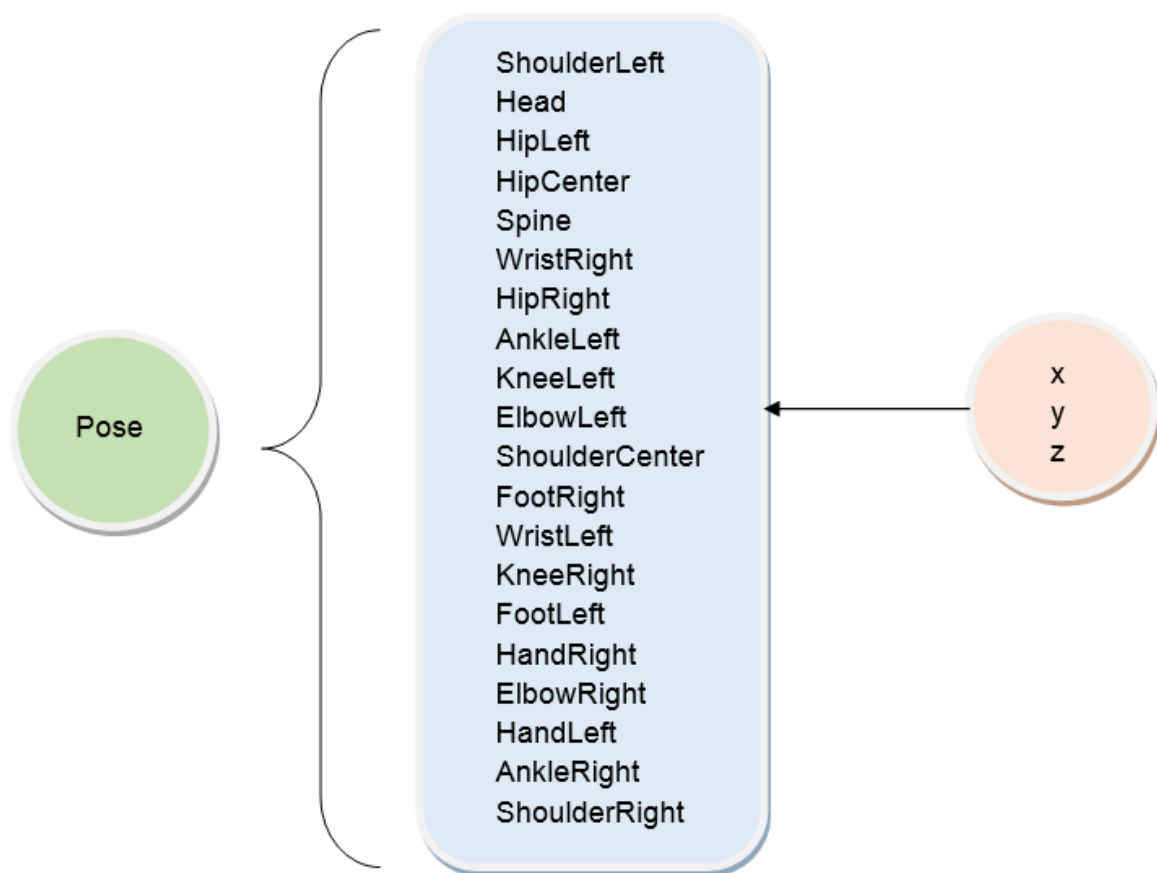
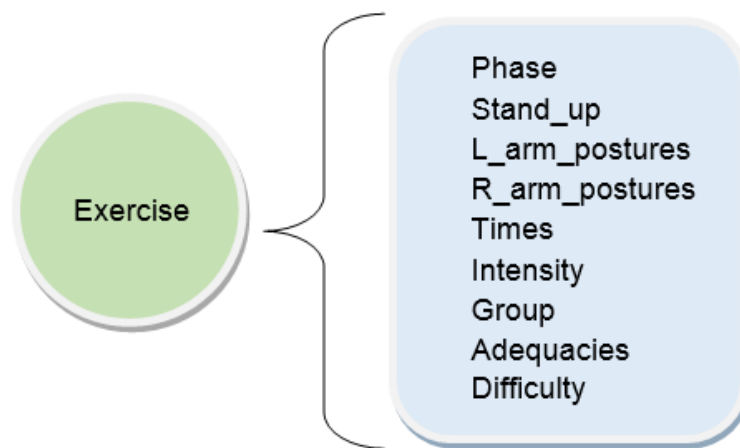


Figura 62. Estructura de ChainsJoint.

#### 3.2.7.4.- Exercises

Este fichero se encarga de almacenar los ejercicios para que el componente de planning pueda utilizarlos. Un ejercicio está compuesto por: la fase a la que pertenece, estado que indica si está de pie o sentado (true/false), las poses del brazo izquierdo y las del brazo derecho, así como el tiempo que mantiene dichas poses, la intensidad del ejercicio, el grupo de ejercicios al que pertenece, los grados de adecuación del ejercicio y la dificultad del mismo (Figura 63). Todos los datos son indicados por el médico rehabilitador, salvo los que conciernen a las poses de los brazos y los tiempos que mantienen dichas posiciones.



*Figura 63. Estructura del fichero Exercises de la base de conocimiento.*

## 4.- Evaluación

Para realizar las pruebas del trabajo desarrollado en este proyecto fin de grado se realizarán las pruebas correspondientes a los escenarios mencionados en el apartado 3.1.1.- Escenarios y a los casos de uso del apartado 3.1.2.- Casos de Uso.

### 4.1.- Sistema de pruebas

Para el correcto funcionamiento de los programas desarrollados en este proyecto, se detallarán los requisitos del sistema de pruebas utilizado.

- Procesador: Intel i7-4510U (2 núcleos x 2.0~3.1GHz)
- Memoria RAM: 8Gb
- Disco Duro: 512Gb SSD
- SO: Windows 8.1 x64
- python 2.7.8 x86
- Ice 3.5.1-2 x86 compilado para python 2.7
- Visual Studio Professional 2012
- Kinect for Windows SDK v1.8
- KinectDeveloperToolkit v1.8.0
- Kinect for Windows Language Packs v1.0
- PyQt 4 for python 2.7 x86
- Qt v4
- pyglet 1.1.4

Se añadieron también los siguientes programas para la interacción del sistema con el robot Nao:

- Choregraphe 1.14.5
- pynaoqi 1.14.5 para Python 2.7

Los programas Visual Studio, Kinect Developer Toolkit y Kinect for Windows son necesarios para conectar la Kinect y ejecutar el componente WinKinectComp, por lo que este componente solo funcionaría bajo un sistema operativo Windows. El resto de componentes pueden ser ejecutados en cualquier sistema operativo, ya que la comunicación entre los componentes se realiza mediante el lenguaje Ice y los demás componentes, a excepción de WinKinectComp, están desarrollados en lenguajes que pueden ser ejecutados en cualquier sistema operativo.

## 4.2.- Pruebas

Los componentes realizados en este proyecto fin de grado han sido el componente VisionComp (3.2.5.- VisionComp) y el componente LearningComp (3.2.6.- LearningComp) para ambos componentes se han realizado una serie de pruebas para comprobar que éstos cumplen con los objetivos del proyecto fin de grado (1.2.- Objetivos), con los escenarios (3.1.1.- Escenarios) y sus casos de uso (3.1.2.- Casos de Uso), y con sus características (3.2.5.- VisionComp y 3.2.6.- LearningComp).

### 4.2.1.- Pruebas del componente VisionComp

Para una correcta verificación del funcionamiento del componente VisionComp se han realizado las siguientes pruebas:

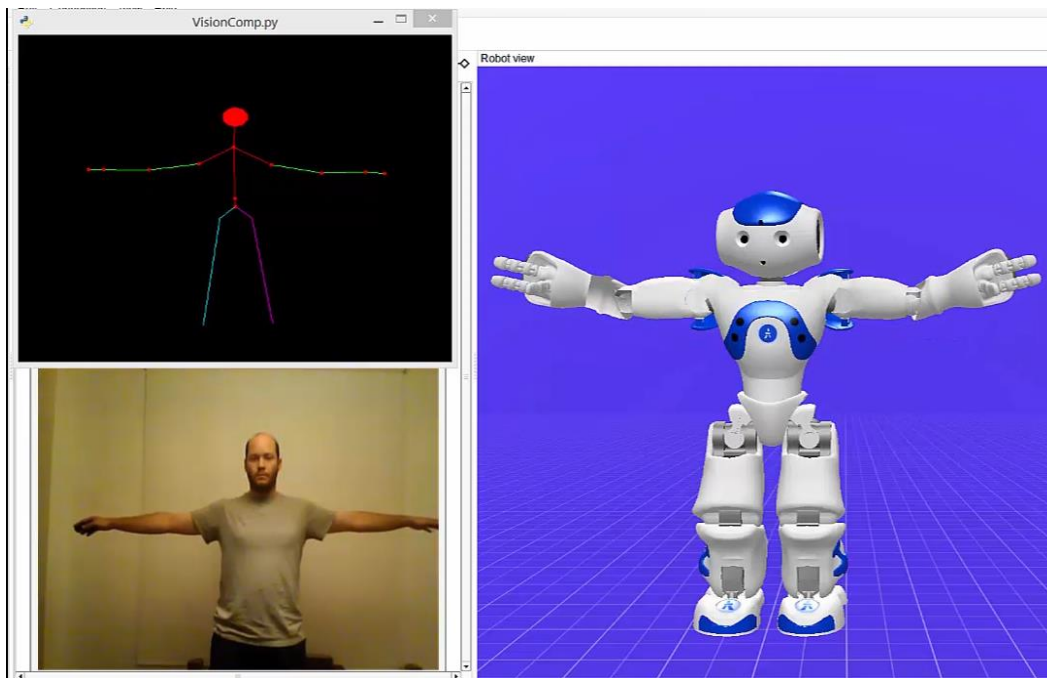
- Comprobar el correcto funcionamiento del componente.
- Comprobar la veracidad de los ángulos que éste calcula.

#### 4.2.1.1.- *Funcionamiento del componente*

Para verificar el correcto funcionamiento del componente se ha enlazado directamente éste con el componente de ejecución del Robot NAO para verificar que los movimientos que realiza el paciente son representados correctamente por el robot NAO y que el visor del esqueleto del componente VisionComp realiza una reconstrucción verídica del mismo (Figura 64).

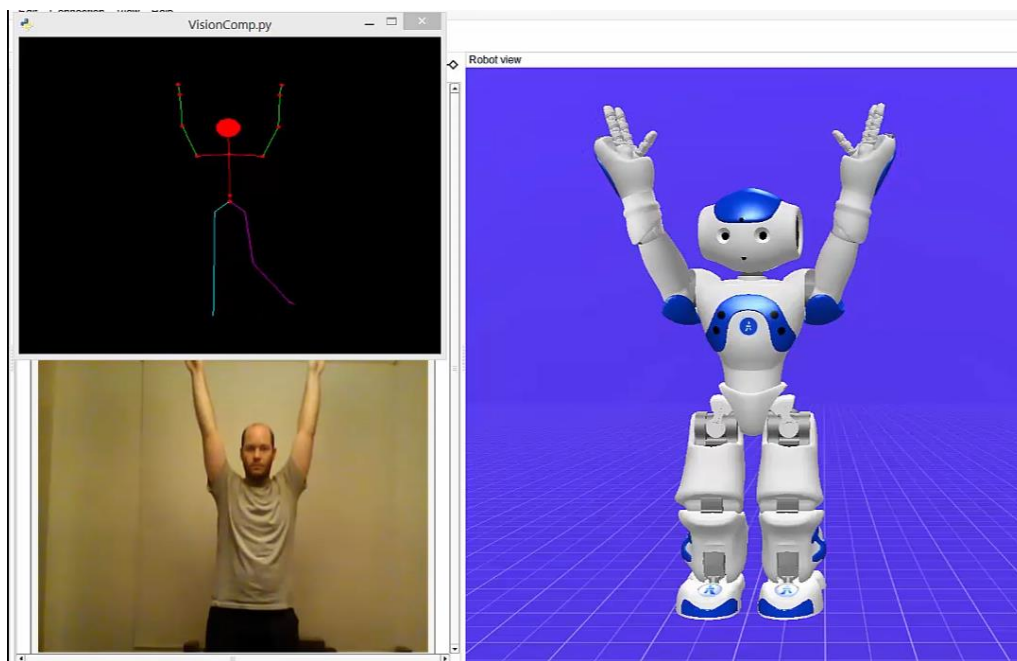
Se puede ver el video completo de la prueba en el Canal de YouTube del proyecto NAOTherapist a través del siguiente enlace:

<http://youtu.be/3F9Hu42ev08>



*Figura 64. Prueba 1 de VisionComp: Postura 1.*

Tal y como se observa en el video y en la Figura 64 y Figura 65, se verifica que la postura que adopta el usuario, es recogida correctamente por el componente VisionComp, el cuál realiza una reconstrucción del esqueleto del usuario y envía los datos al componente de Ejecución que los trasfiere al componente específico Robot NAO, el cuál ordena al robot NAO a adoptar la misma postura que el usuario.



*Figura 65. Prueba 1 de VisionComp: Postura 2.*



#### 4.2.1.2.- Comprobación de los ángulos.

Para realizar esta prueba del componente VisionComp se tuvo que enlazar directamente este componente con el componente de ejecución del Robot NAO para que a través de la interfaz del Robot NAO poder comprobar que los ángulos que se estaban recogiendo del paciente eran correctos. (Figura 66 y Figura 67).

Se puede ver el video completo de la prueba en el Canal de YouTube del proyecto NAOTherapist a través del siguiente enlace:

<http://youtu.be/QJx-XBNPQ0A>

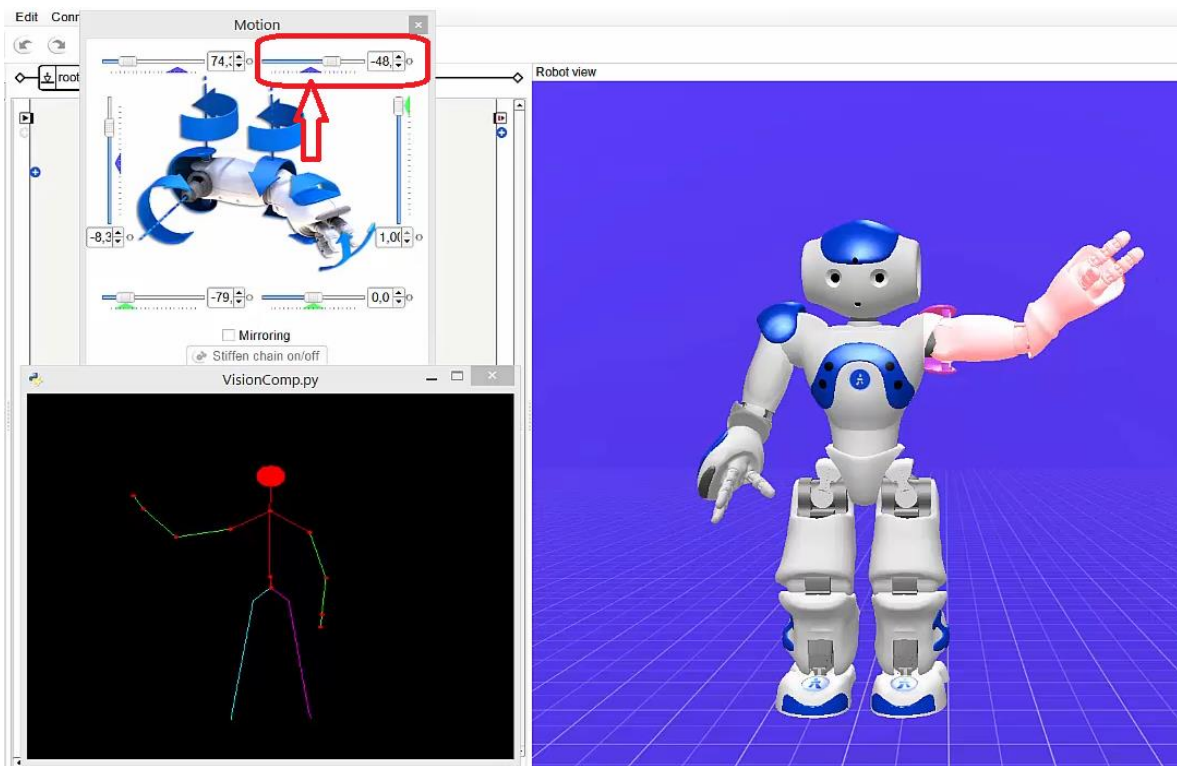


Figura 66. Prueba 2 de VisionComp: Ángulo 45°.

Tal y como se observa en el video y en la Figura 66 y Figura 67, se puede comprobar que el ángulo que adopta el robot NAO es el mismo que adopta el usuario y el visor del esqueleto.

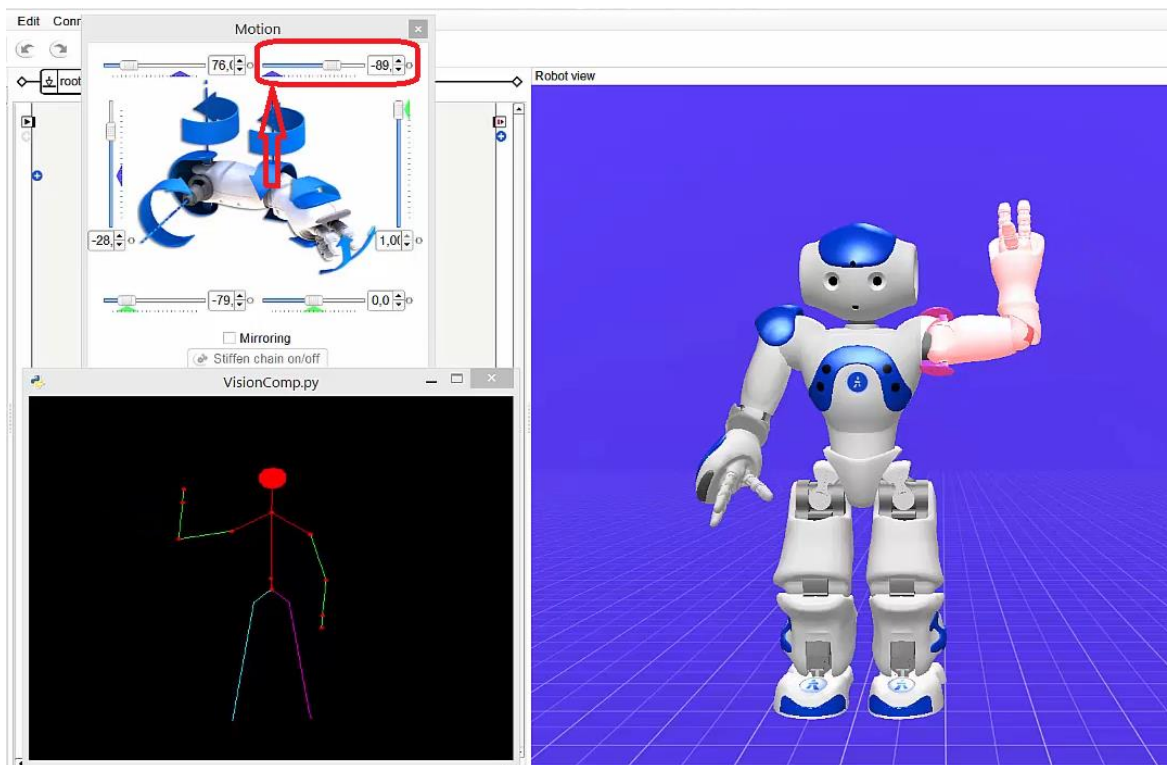


Figura 67. Prueba 2 de VisionComp: Ángulo 90°.

#### 4.2.2.- Pruebas del componente LearningComp

Para una correcta verificación del funcionamiento del componente LearningComp se han realizado las siguientes pruebas:

- Comprobar el correcto funcionamiento del componente a la hora de grabar una pose.
- Comprobar el correcto funcionamiento del componente a la hora de grabar un ejercicio.

Cabe indicar que para la interacción con la interfaz es necesario disponer de un ratón inalámbrico o de otro operario para poder presionar los botones mientras se realizan las poses.

##### 4.2.2.1.- Grabar una pose

Para verificar esta prueba se ha realizado todos los pasos necesarios para grabar una pose, tal y como se indican en el apartado 3.2.6.2.- Captura de poses. Al presionar el botón “Capture Pose” se muestra una cuenta desde 1 hasta 3 y después se realiza la captura (Figura 68).

Se puede ver el video completo de la prueba en el Canal de YouTube del proyecto NAOTherapist a través del siguiente enlace:

<http://youtu.be/sCdLh86TATg>

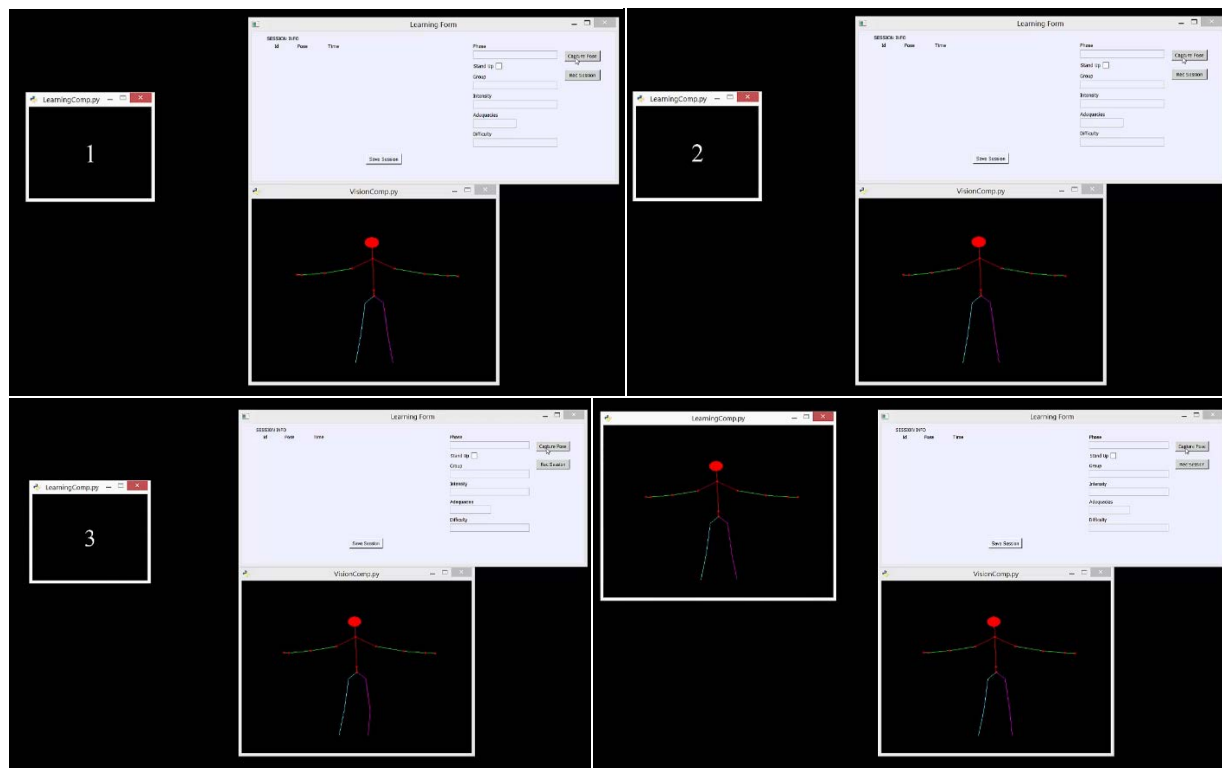


Figura 68. Prueba 1 de LearningComp: Capturar pose.

Esta pose es almacenada en la base de conocimiento según la estructura explicada en el apartado 3.2.7.- Estructura de la base de conocimiento, donde son almacenados cada brazo por separado: ChainsL (Figura 69), ChainsR (Figura 70) y ChainsJoint (Figura 71), en los cuales se verifica que se graba la pose correctamente.

```
<posture id="l1">
  <HandAngleLeft>0.0,1.0</HandAngleLeft>
  <ElbowShouldersRotationLeft>-1.07457387447,-1.07457387447</ElbowShouldersRotationLeft>
  <ElbowShoulderHipRotationLeft>-0.496222496033,-0.496222496033</ElbowShoulderHipRotationLeft>
  <UpperArmCoronalPlaneLeft>-0.234616950154,1.80541324615</UpperArmCoronalPlaneLeft>
  <UpperArmSagittalPlaneLeft>0.496314525604,1.0744818449</UpperArmSagittalPlaneLeft>
  <UpperArmTransversePlaneLeft>-1.01230978966,2.58310604095</UpperArmTransversePlaneLeft>
  <ArmCoronalPlaneLeft>-0.12385250628,1.69464886189</ArmCoronalPlaneLeft>
  <ArmSagittalPlaneLeft>0.499639928341,1.07115638256</ArmSagittalPlaneLeft>
  <ArmTransversePlaneLeft>-1.05321478844,2.62401127815</ArmTransversePlaneLeft>
  <ElbowAngleLeft>-1.34993088245,2.92072725296</ElbowAngleLeft>
  <WristAngleLeft>-1.30439078808,2.87518715858</WristAngleLeft>
</posture>
```

Figura 69. Prueba 1 de LearningComp: ChainsL.

```

<posture id="r1">
  <UpperArmCoronalPlaneRight>-0.124261379242,1.69505774975</UpperArmCoronalPlaneRight>
  <UpperArmSagittalPlaneRight>-0.529646396637,2.10044264793</UpperArmSagittalPlaneRight>
  <UpperArmTransversePlaneRight>-1.02370095253,2.59449720383</UpperArmTransversePlaneRight>
  <ArmCoronalPlaneRight>-0.0555463135242,1.62634265423</ArmCoronalPlaneRight>
  <ArmSagittalPlaneRight>-0.622939646244,2.19373583794</ArmSagittalPlaneRight>
  <ArmTransversePlaneRight>-0.944607853889,2.5154042244</ArmTransversePlaneRight>
  <ElbowAngleRight>-1.34634637833,2.91714286804</ElbowAngleRight>
  <ElbowShouldersRotationRight>0.779040515423,0.779040515423</ElbowShouldersRotationRight>
  <ElbowShoulderHipRotationRight>-0.791755795479,-0.791755795479</ElbowShoulderHipRotationRight>
  <WristAngleRight>-1.23317062855,2.80396699905</WristAngleRight>
  <HandAngleRight>0.0,1.0</HandAngleRight>
</posture>

```

Figura 70. Prueba 1 de LearningComp: ChainsR.

```

<posture id="r1-l1">
  <ShoulderLeft>-0.489352285862,0.0115532176569,1.59806716442</ShoulderLeft>
  <Head>-0.321912825108,0.291913688183,1.61203658581</Head>
  <HipLeft>-0.346679747105,-0.266626358032,1.58119010925</HipLeft>
  <HipCenter>-0.303075999022,-0.190061315894,1.60672175884</HipCenter>
  <Spine>-0.317287117243,-0.141373321414,1.63710188866</Spine>
  <WristRight>0.0368205904961,-0.255167543888,1.69750666618</WristRight>
  <HipRight>-0.237870335579,-0.261072427034,1.61380970478</HipRight>
  <AnkleLeft>-0.510940372944,-0.741300165653,1.63143837452</AnkleLeft>
  <KneeLeft>-0.452433377504,-0.572228968143,1.62042105198</KneeLeft>
  <ElbowLeft>-0.570141434669,-0.142264142632,1.59266757965</ElbowLeft>
  <ShoulderCenter>-0.349135249853,0.127577185631,1.63853168488</ShoulderCenter>
  <FootRight>-0.132250830531,-0.771629035473,1.53446960449</FootRight>
  <WristLeft>-0.642545282841,-0.296461462975,1.54962623119</WristLeft>
  <KneeRight>-0.180320858955,-0.582706451416,1.60344958305</KneeRight>
  <FootLeft>-0.508484244347,-0.764697313309,1.58044350147</FootLeft>
  <HandRight>0.0951209366322,-0.286109805107,1.69420015812</HandRight>
  <ElbowRight>-0.0818613991141,-0.128404840827,1.69402551651</ElbowRight>
  <HandLeft>-0.665030837059,-0.349519342184,1.51727581024</HandLeft>
  <AnkleRight>-0.148482233286,-0.760646879673,1.58703100681</AnkleRight>
  <ShoulderRight>-0.170117914677,0.018104577437,1.6769323349</ShoulderRight>
</posture>

```

Figura 71. Prueba 1 de LearningComp: ChainsJoint.

#### 4.2.2.2.- Grabar un ejercicio

Para realizar esta prueba de grabación de un ejercicio compuesto por varias poses, manteniendo cada una durante un tiempo determinado, se han seguido los pasos indicados en el apartado 3.2.6.3.- Captura de ejercicios. Al presionar el botón “Rec Session” comienza la grabación del ejercicio, el cual se indica parpadeando el botón en rojo. La grabación comenzará en el momento en que se presione por primera vez el botón “Capture pose”, el cual va grabando las poses y los tiempos entre poses que componen el ejercicio. Una vez se da por completado el ejercicio se tendrá que presionar el botón “Stop session” y

se mostrará en el recuadro de la izquierda de la interfaz las poses que componen el ejercicio así como el tiempo que tiene que mantenerse dicha pose. Al lado de cada pose grabada se mostrarán dos botones uno para ver la pose grabada y otro para eliminar dicha pose del ejercicio (Figura 72).

Se puede ver el video completo de la prueba en el Canal de YouTube del proyecto NAOTherapist a través del siguiente enlace:

<http://youtu.be/9PXPLLXDBWs>



Figura 72. Prueba 2 de LearningComp: Grabar ejercicio.

Como puede observarse en la base de conocimiento del sistema, en el archivo Exercises se ha grabado el ejercicio correctamente, por lo que verificamos que el sistema funciona correctamente (Figura 73).

```
<exercise id="e70">
  <phase>warmup</phase>
  <stand_up>true</stand_up>
  <l_arm_postures>p0,p0,p0,p0,p0,p0,p0,p0,p0,p0,p0,p0,p0,p0,p0,p0,p0,p0,p0,p0,p0,p0,p0,p0,p0</l_arm_postures>
  <r_arm_postures>p2,p5,p2,p5,p2,p5,p2,p5,p2,p5,p2,p5,p2,p5,p2,p5,p2,p5,p2,p5,p2,p5,p2,p5,p2,p5,p2,p5</r_arm_postures>
  <times>1,2,4,6,8,10,12,14,16,18,20,22,24,26,30,34,38,42,46,50,52,52,54,56,58,62,66,70,74,78,80,82,84,86,88,90</times>
  <intensity>35</intensity>
  <group>g_bimanual_dissociation</group>
  <adequacies>2,1,0,0,0</adequacies>
  <difficulty>5</difficulty>
</exercise>
```

Figura 73. Prueba 2 de LearningComp: Exercises.

## 5.- Conclusiones

Este trabajo, al pertenecer al proyecto NAOTherapist, es un proyecto fin de grado que está siendo utilizado en dicho proyecto para llevar a cabo los objetivos del mismo<sup>18</sup>.

Del proyecto NAOTherapist heredó algunas de sus limitaciones/ requisitos que en algunos momentos fueron un impedimento para completar los objetivos planteados en el apartado 1.2.- Objetivos, pero que otras veces ayudaron a simplificar tareas.

Los objetivos principales de este trabajo fin de grado son proporcionar un componente que obtuviese los datos de WinKinectComp y los transformase para introducir el cálculo de ángulos. Para ello, se creó un modelo basado en el que proporcionaba WinKinectComp, pero ampliando la información con los datos que son necesarios para realizar este proyecto fin de grado, así como de ser útil para el proyecto NAOTherapist. Gracias al modelo y a los componentes desarrollados en este trabajo, el proyecto NAOTherapist ha conseguido avanzar hasta tal punto que actualmente se están realizando pruebas con niños en hospitales. Tal y como demuestran las pruebas realizadas, el componente VisionComp es capaz de obtener los datos del componente WinKinectComp, generar un visor del esqueleto, y modificar, ampliar y enviar los datos, por lo que funciona y cumple los objetivos marcados.

Otro de los objetivos de este trabajo es crear una interfaz para que el médico rehabilitador y/o terapeuta puedan introducir nuevos ejercicios y poses a la base de conocimiento. Para ello, se creó el componente LearningComp que se encargaba de proporcionar una interfaz para tal función. Este componente tal y como se muestran en las pruebas realizadas, proporciona una interfaz sencilla, pero funcional, ya que cumple con los objetivos del proyecto.

El éxito de este trabajo y todas sus implicaciones pueden verse reflejadas a través del proyecto NAOTherapist, el cual está teniendo una gran aceptación por parte de los pacientes, los médicos y los medios de comunicación, los cuales han realizado numerosos reportajes sobre el tema.

<https://www.youtube.com/watch?v=bCAvfhrp6JQ>

---

<sup>18</sup> [www.youtube.com/user/NAOTherapist](https://www.youtube.com/user/NAOTherapist)

## **5.1.- Futuros trabajos**

Sería interesante para futuros trabajos, conseguir una imagen en color y de profundidad de lo que capta el sensor 3D Kinect, así como poder dar órdenes verbales para no tener una dependencia del ratón o de otro operador a la hora de interactuar con la interfaz del componente LearningComp.

Otra futura mejora podría ser realizar modificaciones en la interfaz gráfica del componente LearningComp (3.2.6.1.- Interfaz gráfica) para que fuese más agradable visualmente, así como la transformación de los datos de la cara y las manos del modelo VisionComp (3.2.4.- Modelo de VisionComp) que han sido introducidos del mismo modo que son recibidos del componente WinKinectComp y que no son utilizados en este trabajo por lo que no se realiza ninguna modificación sobre ellos.

## 6.- Referencias y Bibliografía

- [Alfaro, 2012] Alfaro Ballesteros, S. (2012). *Proyecto fin de carrera*. Universidad Carlos III de Madrid. (Citada en pág. 23)
- [Bednarik & Herman, 2015] Bednarik, J., & Herman, D. (2015). Human gesture recognition using top view depth. *Excel@FIT*. Brno. (Citada en pág. 28)
- [Calderita, 2013] Calderita Estévez, L. V., Bustos García de Castro, P., Suárez Mejías, C., Fernández Rebollo, F., & Bandera Rubio, A. (2013). THERAPIST: towards an autonomous socially interactive robot for motor and neurorehabilitation therapies for children. *PervasiveHealth '13 Proceedings of the 7th International Conference on Pervasive Computing Technologies for Healthcare*, (págs. 374-377). (Citada en pág. 10)
- [Castelli, 2011] Castelli, E. (2011). Robotic movement therapy in cerebral palsy. *Developmental Medicine & Child Neurology*, pág. 481. (Citada en pág. 10)
- [Fong & Nourbakhsh, 2013] Fong, T., & Nourbakhsh, I. (2013). *Robotics and Autonomous Systems* (Vol. 42 ). Lausanne, Switzerland. (Citada en pág. 13)
- [Fraser *et al*, 2013] Fraser, A., Tovi, G., Justin, M., & George, F. (2013). YouMove: Enhancing Movement Training with an Augmented Reality Mirror. *UIST'13*. (Citada en pág. 20)
- [González, 2014] González, J. C. (2014). Integración de Planificación de Alto Nivel, Ejecución y Percepción en Robots Autónomos. *Tesis de fin de máster*. Universidad Carlos III de Madrid. (Citada en pág. 9)
- [González-Pacheco *et al*, 2013] Gonzalez-Pacheco, V., Malfaz, M., Fernandez, F., & Salichs, M. A. (2013). Teaching Human Poses Interactively to a Social Robot. (13, 12406-12430;). *Sensors*. (Citada en pág. 27)
- [Li *et al*, 2013] Li, B., Bentley, C. U., WA, A., Mian, A., Liu, W., & Krishna, A. (2013). Using Kinect for face recognition under varying poses, expressions, illumination and disguise. *Applications of Computer Vision (WACV), 2013 IEEE Workshop on*. Tampa, FL. (Citada en pág. 33)
- [Lui *et al*, 2010] Lui, Y. M., Dept. of Comput. Sci., C. S., Beveridge, J., & Kirby, M. (2010). Action classification on product manifolds. *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, (págs. 833 - 839). San Francisco, CA. (Citada en pág. 27)
- [Manso *et al*, 2010] Manso, L., Bachiller, P., Bustos, P., Nuñez, P., Cintas, R., & Calderita, L. (2010). RoboComp: a Tool-based Robotics Framework. *SIMPAR Second International Conference on Simulation, Modeling and Programming for Autonomous Robots*, (págs. 251-262). (Citada en pág. 17)
- [Mariusz *et al*, 2013] Mariusz, D., Wojciech, R., Slawomir, S., Joanna, D., Magdalena, S., Ewelina, Z., Grzegorz, S. (2013). Functional effects of robotic-assisted locomotor treadmill therapy in children with cerebral palsy. *J Rehabil Med*, 45: 358–363. (Citada en pág. 10)
- [O'Hara *et al*, 2012] O'Hara, S., Lui, Y. M., & Draper, B. A. (2012). Using a Product Manifold distance for unsupervised action recognition. *Image and Vision Computing*, 30, 206-216. (Citada en pág. 27)
- [Pulido, 2014] Pulido, J. C. (2014). Planificación Jerárquica y Detección de Poses para el Desarrollo de Terapias de Rehabilitación con Robots Sociales. Universidad Carlos III de Madrid. (Citada en pág. 9)

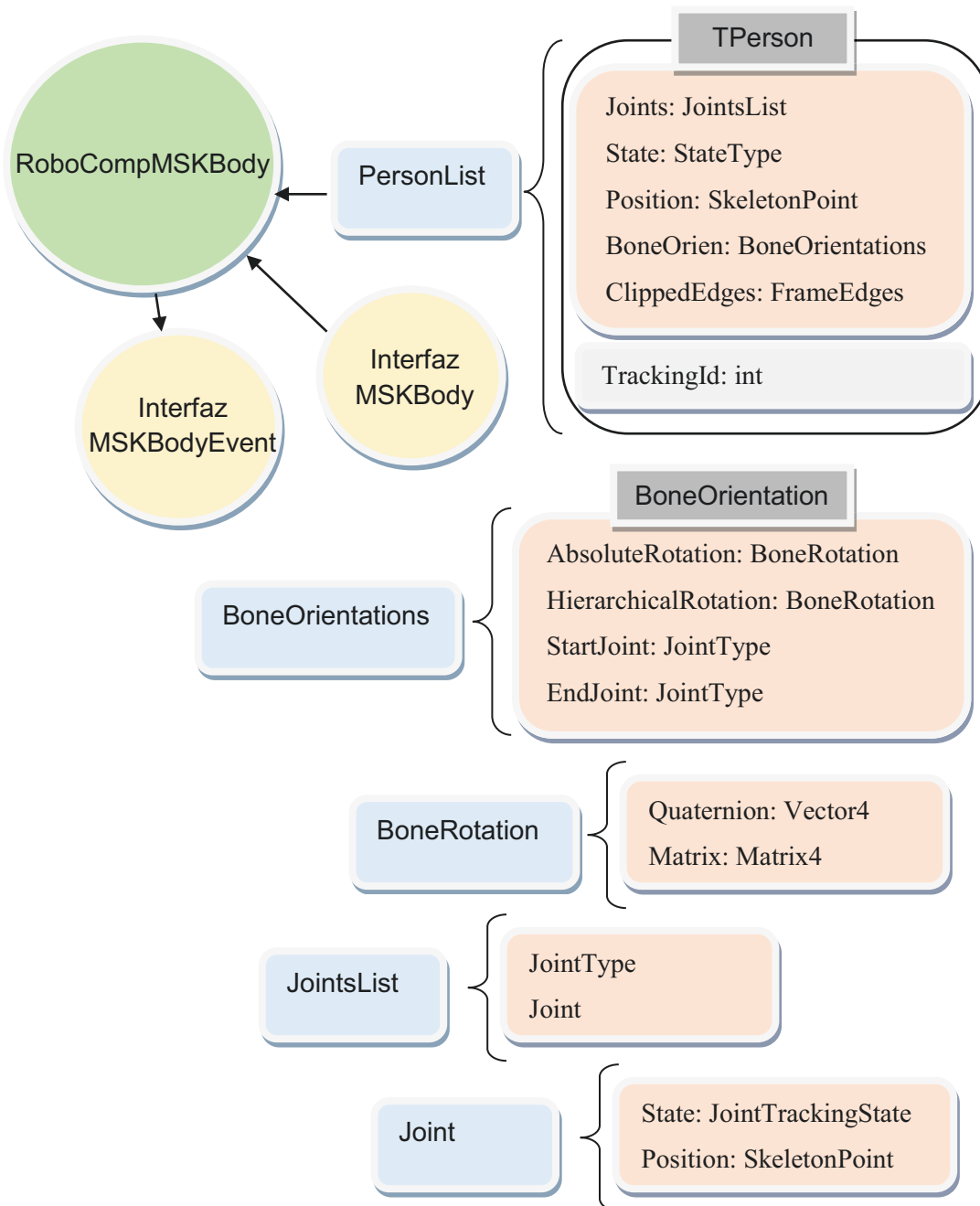


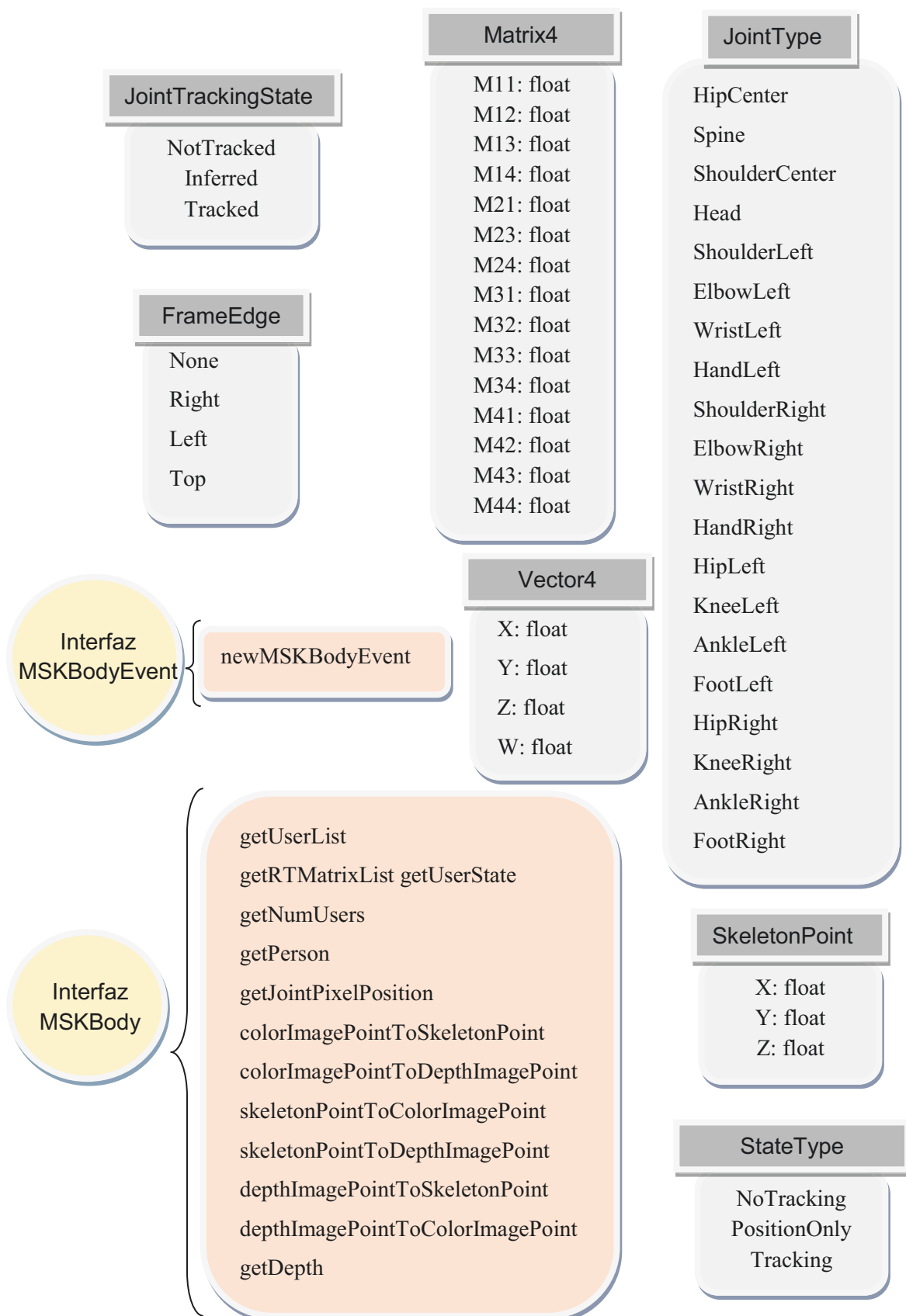
- [Pulido *et al*, 2014] Pulido, J. C., González, J. C., González, A., García, J., Fernández, F., Bandera, A., Suárez, C. (2014). Goal-directed Generation of Exercise Sets for Upper-Limb Rehabilitation. *Proceedings of the 5th Workshop on Knowledge Engineering for Planning and Scheduling (KEPS)*, (págs. 38-45). Portsmouth (New Hampshire, USA). (Citada en pág. 9)
- [Raheja *et al*] Raheja, J., Group, D. S., Lab., M. V., Pilani, I., Chaudhary, A., & Singal, K. (2011). Tracking of Fingertips and Centers of Palm Using KINECT. *Computational Intelligence, Modelling and Simulation (CIMSIM), 2011 Third International Conference on*. Langkawi. (Citada en pág. 32)
- [Raquel *et al*, 2011] Raquel, R., Marco, N., Rachel, W., Paul, B., Rosemarijn, L., & Yiannis, D. (2011). Child-Robot Interaction in The Wild: Advice to the Aspiring. *Proceedings of the ACM International Conference on Multi-modal interaction*, (págs. 335-342). Valencia, Spain. (Citada en pág. 11)
- [Won *et al*, 2014] Won, A. S., Bailenson, J. N., Stathatos, S. C., & Dai, W. (22 de Mayo de 2014). Automatically Detected Nonverbal Behavior Predicts. (Science Business). Springer. (Citada en pág. 13)
- [García *et al*, 2011] E. García, D. Sanz-Merodio, F. Sanchez, J.C. Arévalo and P. González de Santos (2011). Development of the ATLAS lower-limb active orthosis, 14th Int. Conf. Climbing and Walking Robots and the Support Technologies for Mobile Machines , Paris, France, 2011. (Citada en pág. 13)

## 7.- Anexo A: Modelos

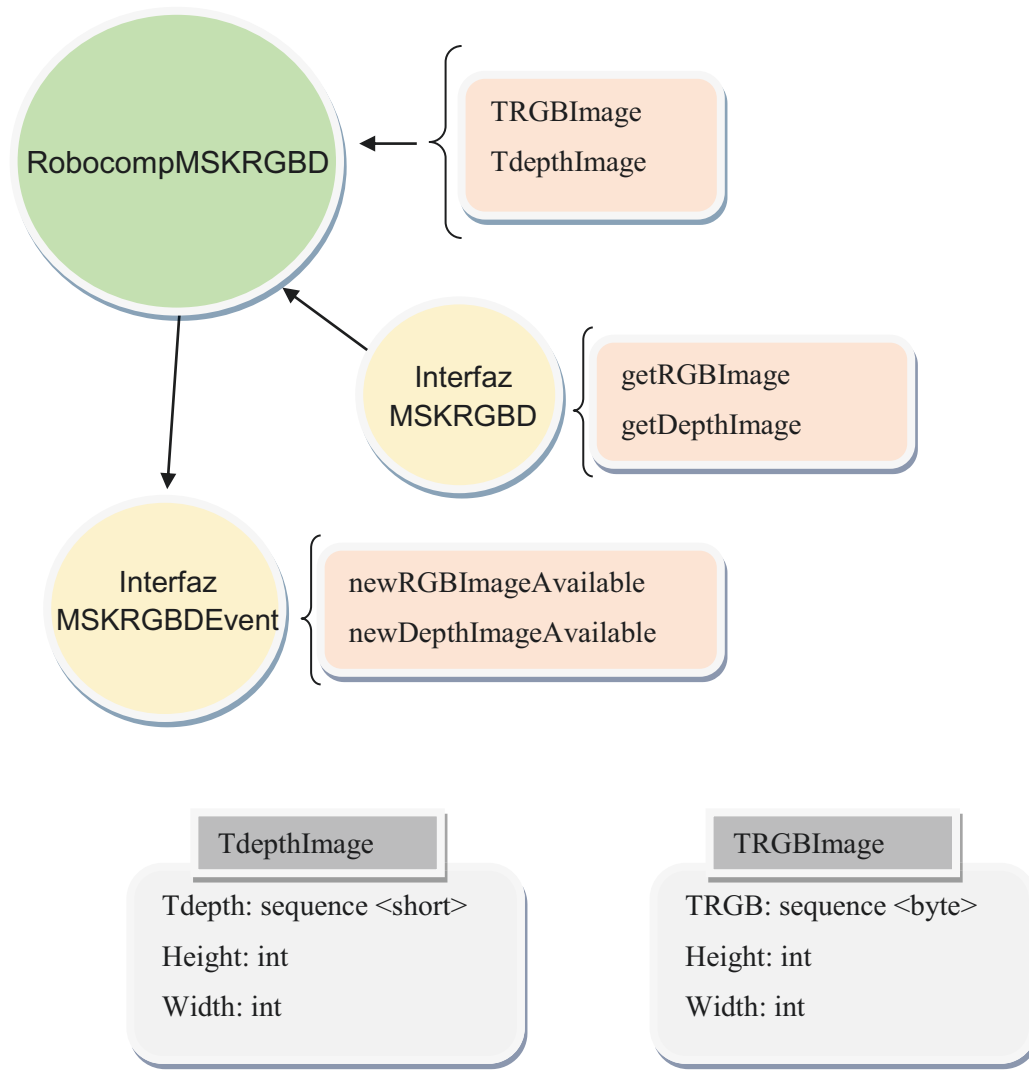
### 7.1.- Modelo de WinKinectComp

#### 7.1.1.- Módulo RoboCompMSKBody

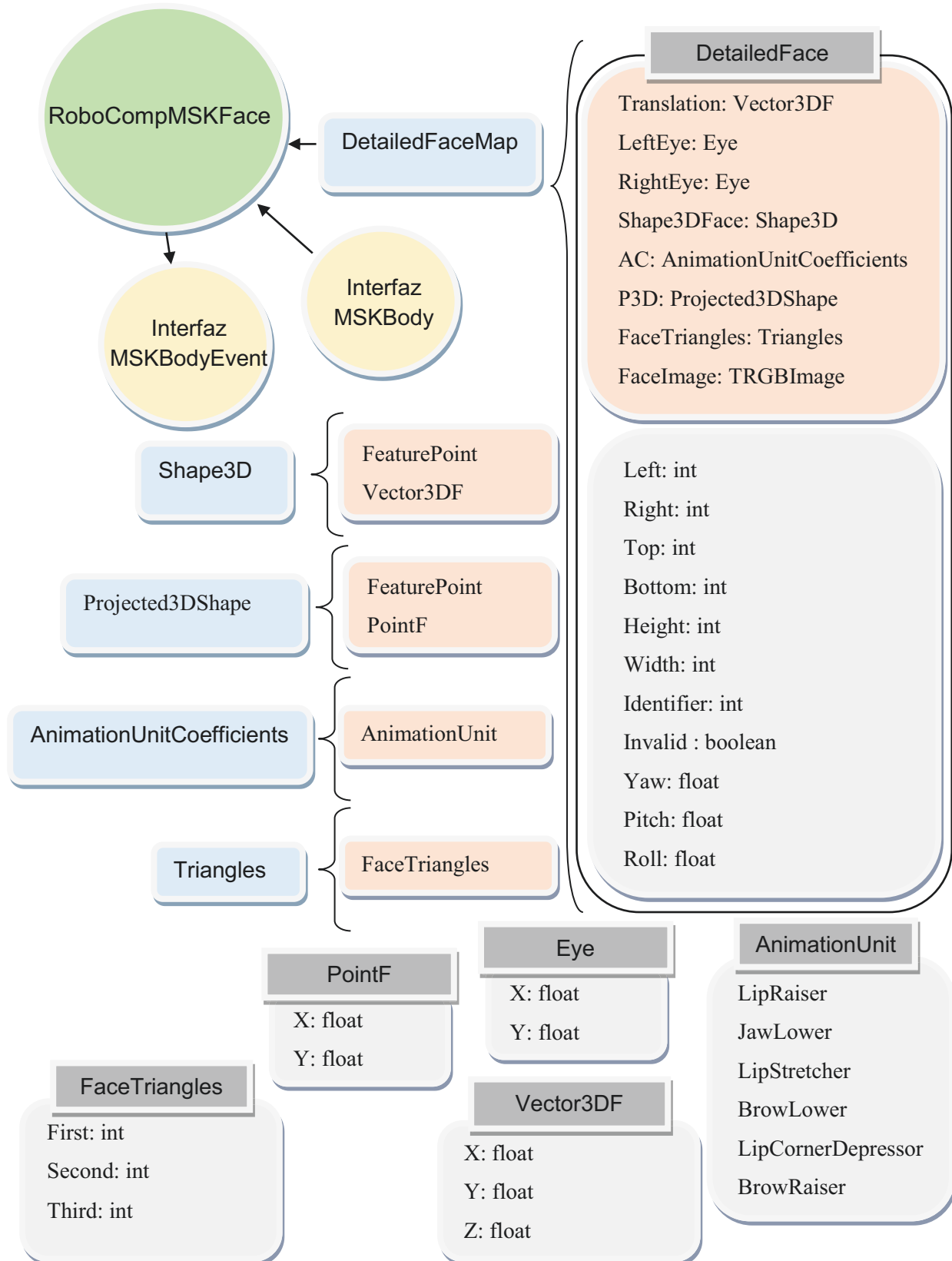




### 7.1.2.- Módulo RobocompMSKRGBD

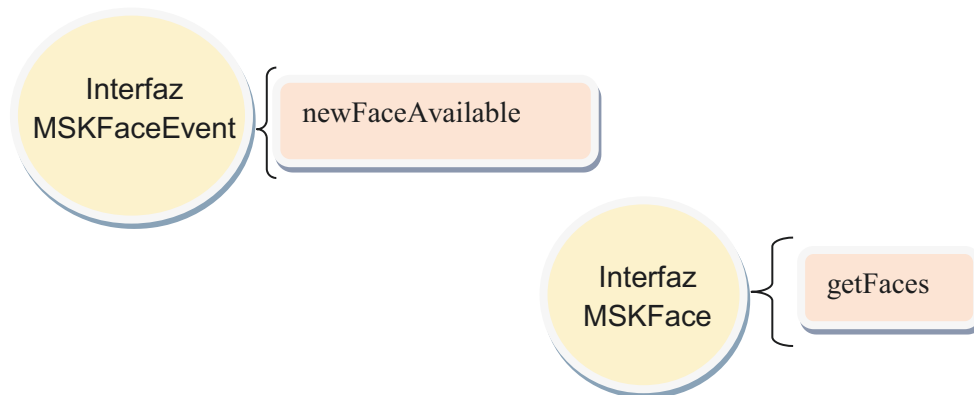


### 7.1.3.- Módulo RoboCompMSKFace

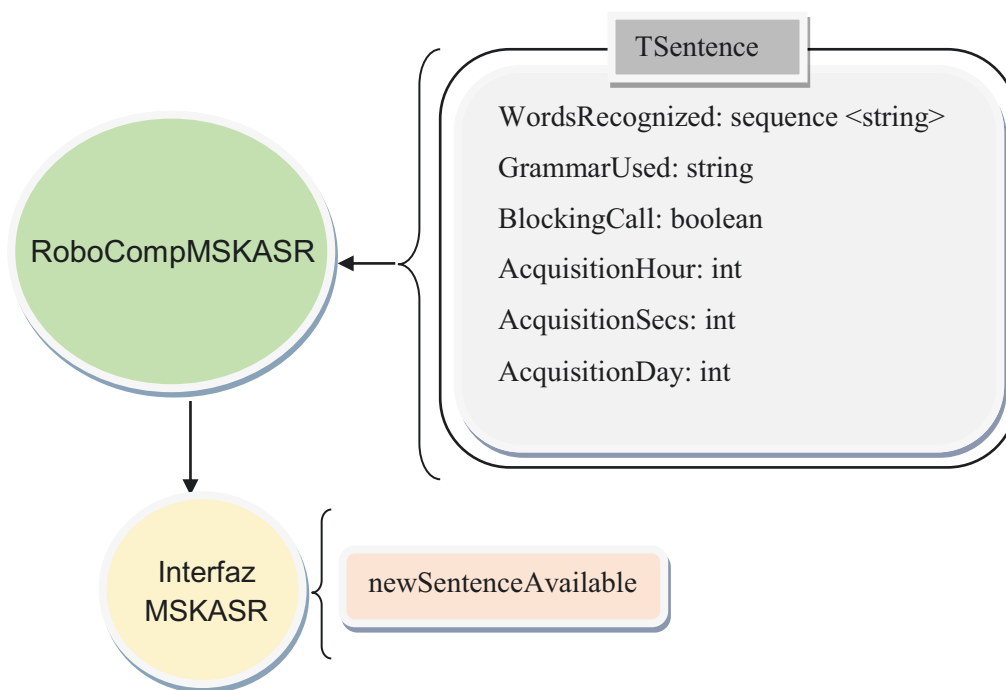


## FeaturePoint

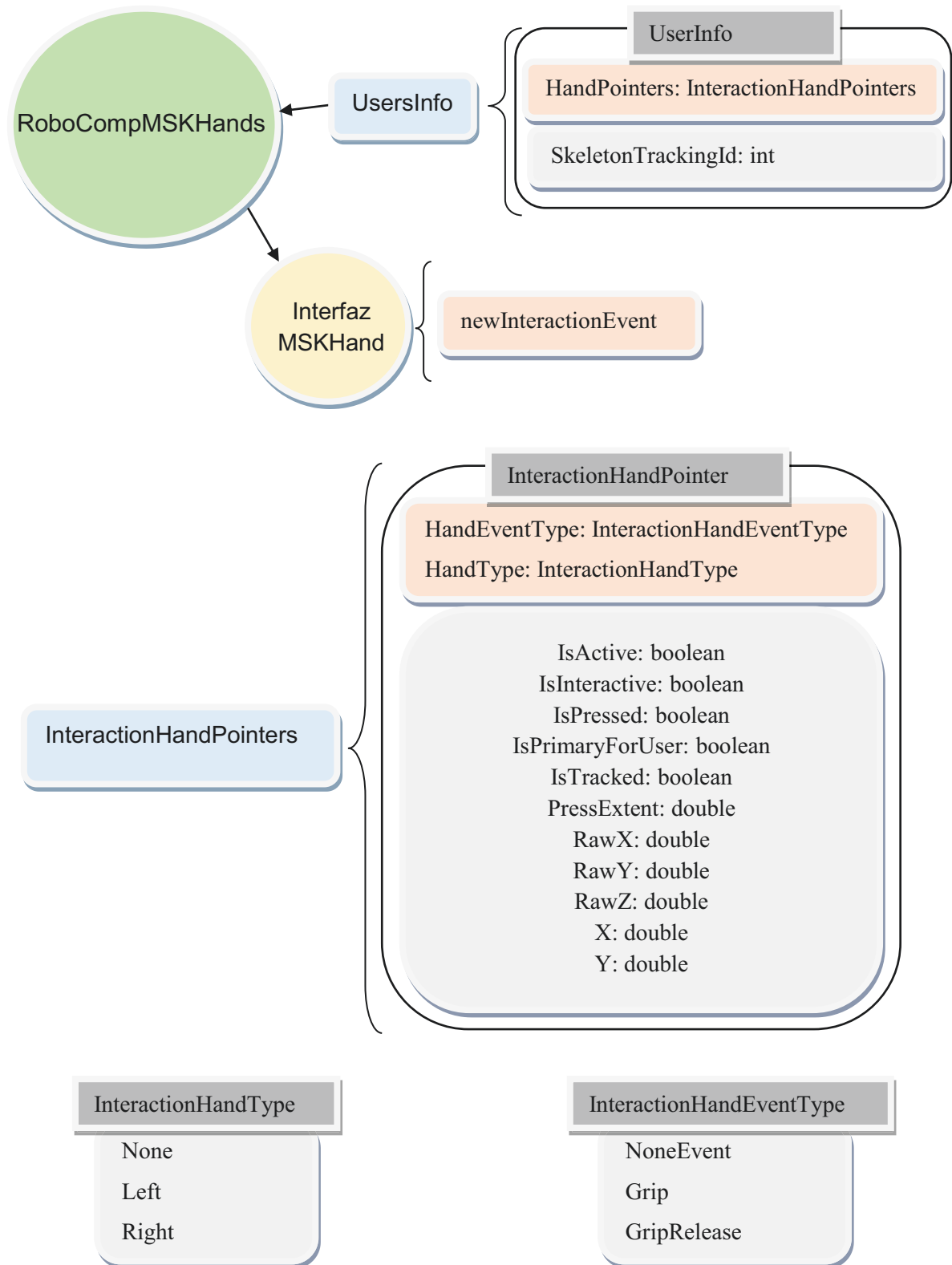
TopSkull	FP42	RightBottomLowerLip
TopRightForehead	FP43	LeftBottomLowerLip
FP2	FP44	MiddleBottomUpperLip
FP3	FP45	LeftCornerMouth
FP4	FP46	RightCornerMouth
FP5	FP47	BottomOfRightCheek
FP6	LeftOfLeftEyebrow	BottomOfLeftCheek
MiddleTopDipUpperLip	MiddleTopOfLeftEyebrow	FP92
F8	RightOfLeftEyebrow	FP93
AboveChin	MiddleBottomOfLeftEyebrow	FP94
BottomOfChin	AboveMidUpperLeftEyelid	AboveThreeFourthRightEyelid
F11	OuterCornerOfLeftEye	AboveThreeFourthLeftEyelid
FP12	MiddleTopLeftEyelid	ThreeFourthTopRightEyelid
FP13	MiddleBottomLeftEyelid	ThreeFourthTopLeftEyelid
FP14	InnerCornerLeftEye	ThreeFourthBottomRightEyelid
RightOfRightEyebrow	UnderMidBottomLeftEyelid	ThreeFourthBottomLeftEyelid
MiddleTopOfRightEyebrow	FP58	BelowThreeFourthRightEyelid
LeftOfRightEyebrow	FP59	BelowThreeFourthLeftEyelid
MiddleBottomOfRightEyebrow	FP60	AboveOneFourthRightEyelid
AboveMidUpperRightEyelid	FP61	AboveOneFourthLeftEyelid
OuterCornerOfRightEye	FP62	OneFourthTopRightEyelid
MiddleTopRightEyelid	LeftSideOfCheek	OneFourthTopLeftEyelid
MiddleBottomRightEyelid	OutsideLeftCornerMouth	OneFourthBottomRightEyelid
InnerCornerRightEye	LeftOfChin	OneFourthBottomLeftEyelid
UnderMidBottomRightEyelid	LeftTopDipUpperLip	FP109
FP25	OuterTopRightPupil	FP110
FP26	OuterBottomRightPupil	FP111
FP27	OuterTopLeftPupil	FP112
FP28	OuterBottomLeftPupil	FP113
FP29	InnerTopRightPupil	FP114
RightSideOfChin	InnerBottomRightPupil	FP115
OutsideRightCornerMouth	InnerTopLeftPupil	FP116
RightOfChin	InnerBottomLeftPupil	FP117
RightTopDipUpperLip	FP75	FP118
TopLeftForehead	FP76	FP119
FP35	FP77	FP120
FP36	FP78	
FP37	RightTopUpperLip	
FP38	LeftTopUpperLip	
FP39	RightBottomUpperLip	
MiddleTopLowerLip	LeftBottomUpperLip	
MiddleBottomLowerLip	RightTopLowerLip	
	LeftTopLowerLip	



#### 7.1.4.- Módulo RoboCompMSKASR



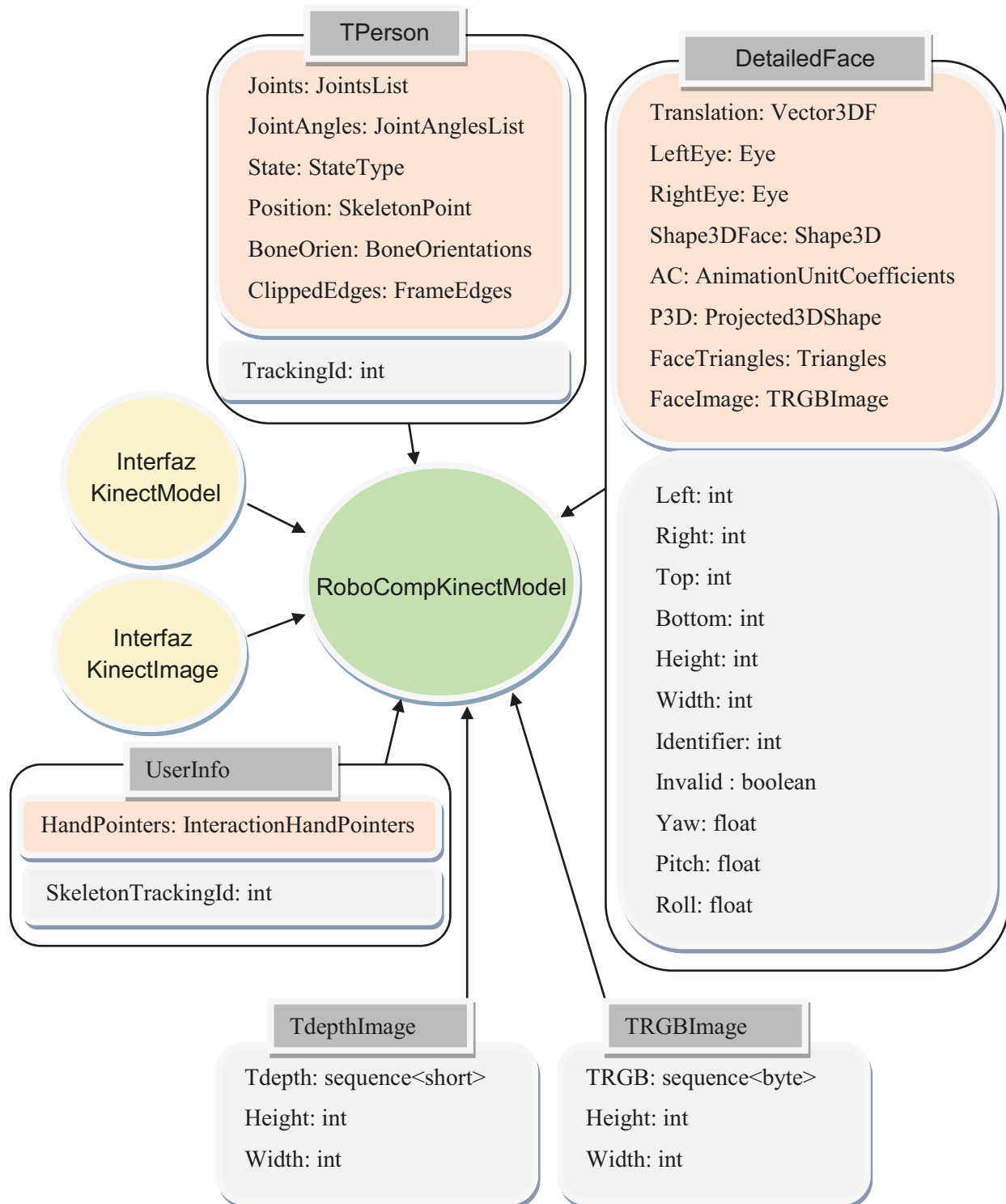
### 7.1.5.- Módulo RoboCompMSKHand



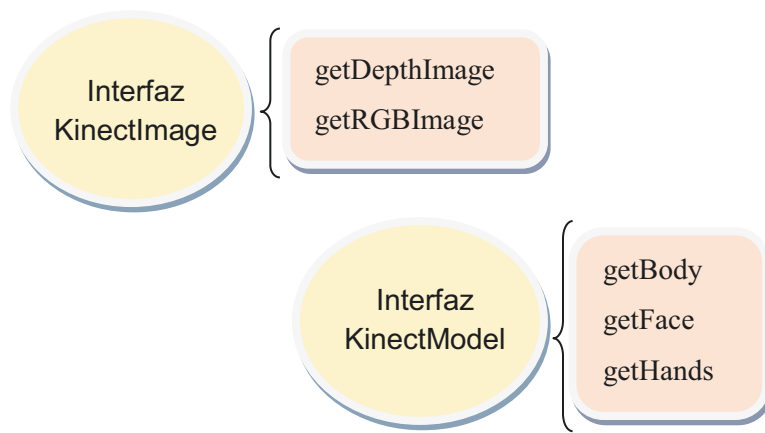


## 7.2.- Modelo de VisionComp

### 7.2.1.- Módulo RoboCompKinectModel







## *8.- Anexo B: Planificación, presupuesto y marco legal*

En este anexo se detallará la planificación realizada durante la elaboración de este proyecto fin de grado, así como el presupuesto calculado para llevarlo a cabo.

### **8.1.- Planificación**

En este apartado se detallará la programación realizada para cada fase del proyecto. Sólo han sido introducidos en la planificación las fases que conformaban un hito dentro del trabajo, para poder obtener una planificación más acorde a la estructura del proyecto, así como para poder cuantificar el tiempo empleado en cada una de las fases. La planificación inicial se propuso que el proyecto se llevaría a cabo a lo largo de 89 días a razón de 3h diarias más 20 días de imprevistos, incluyendo los fines de semana, cuyas horas serían repartidas entre la semana (Figura 74).

Las fases están divididas por hitos importantes del proyecto: comunicación con el componente WinKinectComp, realizar teleoperación con la Kinect y el robot NAO, creación del modelo VisionComp, creación de la interfaz LearningComp, captura de poses con el componente LearningComp, captura de ejercicios con el componente LearningComp y la redacción de la memoria del proyecto. A estas fases se debe sumar una última fase en la cual se tiene en cuenta los desvíos de la planificación realizada por temas externos al proyecto, como pueden ser días de descanso, enfermedad, etc...

La realización final del proyecto supuso un sobrecoste de 59 días, casi dos meses, debido a los problemas encontrados en el desarrollo del proyecto, así como a otros imprevistos (Figura 75). Aunque algunas fases se mantuvieron como fueron planificadas inicialmente, la mayoría de las fases de desarrollo han llevado más tiempo del estimado. (Figura 76).

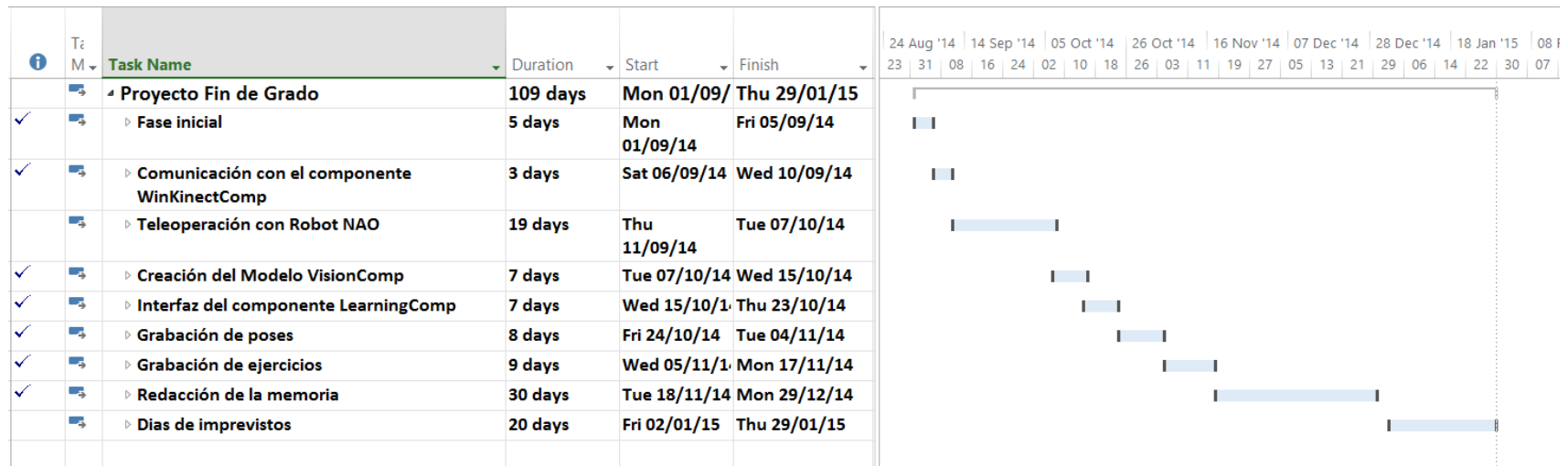


Figura 74. Planificación inicial.

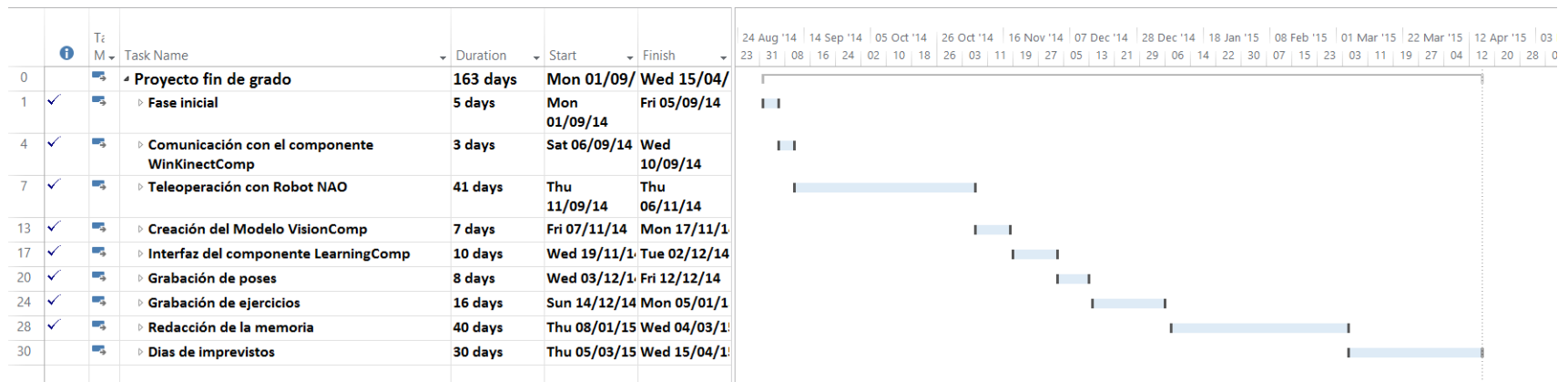


Figura 75. Planificación final.

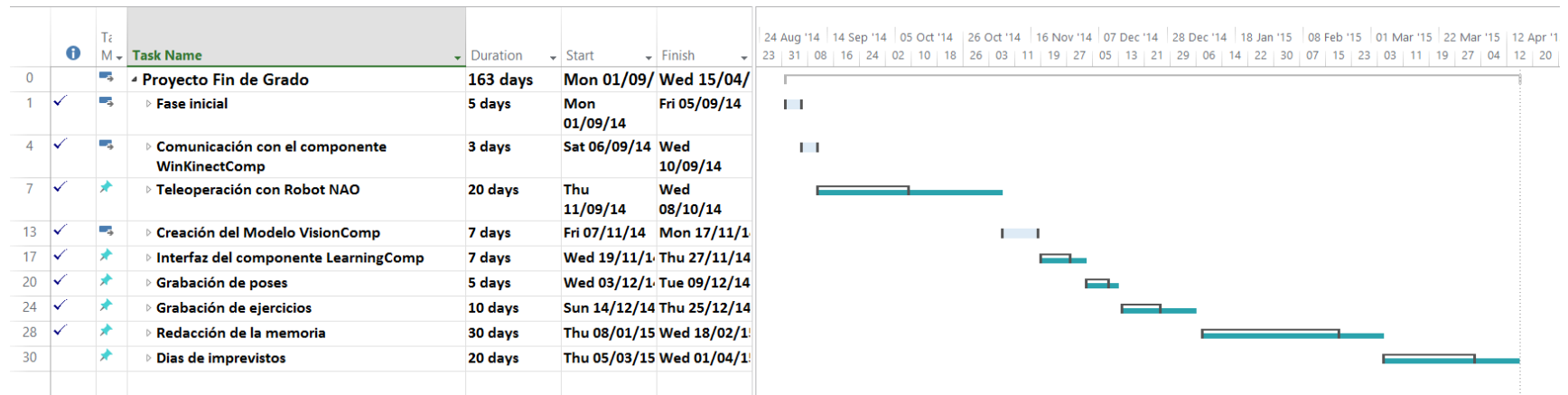


Figura 76. Planificación inicial vs planificación final.

## 8.2.- Presupuesto

En esta sección se detallan los costes producidos por este proyecto y el presupuesto necesario para llevarlo a cabo.

**Autor del presupuesto:** Adrián Rossignoli Martínez-Vara de Rey

**Departamento:** Departamento de PLG de la Universidad Carlos III de Madrid.

Descripción del proyecto:

- **Título:** Desarrollo de terapias de rehabilitación motora teleoperadas con el robot NAO.
- **Duración:** 4 meses.

Presupuesto total del proyecto: 11000€

PERSONAL				
Nombre y Apellidos	Categoría	Horas Trabajadas (H)	Coste/ Hora (€/H)	Coste Total (€)
Adrián Rossignoli	Investigador Principal	140	20,00	2800
Adrián Rossignoli	Desarrollador	349	15,00	5235
TOTAL (€)				8035

*Tabla 1. Presupuesto del personal.*

Si tenemos en cuenta que se ha tardado 5 meses y medio en realizar el trabajo, correspondería a un sueldo de unos 1607€ al mes.

EQUIPOS				
Descripción	Coste (€)	Uso en el proyecto (%)	Periodo de depreciación (meses)	Coste imputable * (€)
PC - Portátil	1000	100	36	80
Robot NAO	5900	10	60	100
* En el coste imputable se ha considerado que los equipos no se compraron para este proyecto.			TOTAL (€)	180

*Tabla 2. Presupuesto del equipo.*

SOFTWARE				
Descripción	Coste (€)	Uso en el proyecto (%)	Periodo de depreciación (meses)	Coste imputable *
Windows 8.1	115	100	30	4
Microsoft Office 2013	80	100	30	3
Aldebaran NAOqi Choregraphe	Free	10	30	0
Pyglet	Free	60	30	0
Notepad ++	Free	80	30	0
Visual Studio Professional 2012	Free	60	30	0
PyQT	Free	40	30	0
* En el coste imputable se ha considerado que los equipos no se compraron para este proyecto.			TOTAL (€)	7

Tabla 3. Presupuesto del software.

OTROS COSTES DEL PROYECTO			
Descripción	Empresa	Coste por mes (4 meses)	Coste imputable *
Internet	Orange	24 * 4 = 96	40
Electricidad	Iberdrola	100 * 4 = 400	150
* En el coste imputable se ha considerado que los equipos no se compraron para este proyecto.			TOTAL (€)
			190

Tabla 4. Presupuesto de otros gastos.



RESUMEN DE COSTES DEL PROYECTO	
Concepto	Coste (€)
Personal	8035
Equipos	180
Software	7
Otros gastos	190
<b>Total sin IVA</b>	8412
<b>IVA (21%)</b>	1766,52
<b>Total con IVA</b>	10178,52

*Tabla 5. Resumen del presupuesto.*

### 8.3.- Marco legal

El contenido de este trabajo fin de grado está protegido por la Ley de Propiedad Intelectual (Real Decreto Legislativo 1/1996, 12 abril), y en su caso por la Ley de patentes (Ley 11/1986, de 20 de marzo, de Patentes). Al tratarse de un trabajo realizado con un robot terapeuta está sujeto a la norma ISO-13482 (2014).